Mining of Ship Operation Data for Energy Conservation

Petersen, Jóan Petur

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Mining of Ship Operation Data for Energy Conservation

Jóan Petur Petersen

Kongens Lyngby 2011
IMM-PHD-2011-XX

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Summary

This thesis presents two state-of-the-art systems approaches to statistical modelling of fuel efficiency in ship propulsion: a regression model and a dynamical model.

Three statistical regression model approaches are investigated and compared: Artificial Neural Networks (ANN), Gaussian processes (GP), and Gaussian Mixture Models (GMM).

A dynamical modelling approach is introduced. This modelling approach has not been used before in the context of ship propulsion modelling, and solves problems encountered with the regression model in an onboard trim optimization application. The dynamical model is introduces through a study of the wellknown sunspot time series, and then on ship data. The dynamical modelling approach is investigated using both the Artificial Neural Network and the Gaussian mixture model.

The thesis also presentes a novel and publicly available data set of high quality sensory data on which all the models are based and tested. No other similar publicly available data set exists. The data presented is a publicly available full-scale data set, with a whole range of features sampled over a period of 2 months. The data is online with an accompanying homepage, where all the results are also presented.

Resumé

Denne afhandling præsenterer to avancerede systemiske tilgange til statistisk modellering af brændstoføkonomi i forbindelse med skibes fremdrift; nemlig en regressionsmodel og en dynamisk model.

Tre forskellige regressionsmodeller undersøges og sammenlignes: Kunstige neurale netværk, Gaussiske processer og Gaussiske miksturmodeller.

Der introduceres en tilgang baseret på dynamisk modellering. Denne tilgang har ikke tidligere været anvendt til modellering af skibes fremdrift. Tilgangen løser en række problemer, der ellers opstår ved regressionsmodellering anvendt til trimoptimering.

Den dynamiske model introduceres gennem en analyse af den velkendte solplettidsserie, samt efterfølgende på skibsdata. Den dynamiske tilgang undersøges både ved anvendelse af kunstige neurale netværk og Gaussiske miksturmodeller.

Afhandlingen præsenterer også et nyt og alment tilgængeligt datasæt af høj kvalitet, hvorpå de nævnte modeller er udviklede og testede. Intet lignende publiceret datasæt eksisterer. Datasættet indeholder fuld-skala sensordata med en lang række signaler, der er målte over en periode på 2 måneder. Det er gjort tilgængeligt online via en tilhørende hjemmeside, hvor alle resultaterne ligeledes præsenteres.

Samandráttur

Henda ritgerð leggur tveir framkomnar hagfrøðiligar myndlar fram, ið meta um orkunýtsluna í sambandi við framdrátt av skipum; nevniliga ein regressiónsmyndil og ein dynamiskan myndil.

Tríggir hættir at gera regressiónsmyndilin verða kannaðir og samanbornir: mannagjørd nervanet, Gaussiskar tilgongdir og Gaussiskir miksturar.

Ein heilt nýggjur háttur, ið byggir á ein dynamiskan myndil, verður lýstur. Hesin hátturin er ikki áður brúktur til at meta um orkutørvin av framdrátti av skipum. Hátturin loysir fleiri av trupulleikunum, ið standast av at brúka regresssiónsmyndilin til at finna tað besta trimmið at sigla við.

Sum innleiðsla til tann dynamiska myndilin, verður hann fyrst brúktur til at rokna talið av sólarblettum á sólini. Síðani verður myndilin víðkaður til eisini at kunna rokna orkunýtsluna av framdrátti av skipum. Sólarblettartrupulleikin er tikin við fyri at gera tað lættari at skilja tann endaliga myndilin til skip. Nógvir trupulleikar, ið eisini standast av at meta um orkunýtsluna hjá skipum, verða tí loystir í einum einfaldari høpi. Tann dynamiski myndilin verður kannaður við nýtslu av mannagjørdum nervalagi og Gaussiskum miksturum.

Ritgerðin lýsir eisini nýggj og almenn atkomilig dáta av høgari góðsku, ið teir áður nevndu myndlarnir eru mentir og royndir á. Einki alment dáta av somu góðsku hevur verið til áður. Forritið til teir mentu myndlarnar og dáta er lýst á heimasíðu, ið er gjørd til endamálið.

Preface

This thesis was prepared at DTU Informatics, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the PhD degree in engineering. The PhD project is done as a collaboration between DTU Informatics and Decision3, a Faroese company.

The thesis deals with statistical modelling of the main propulsion of ships, with special regard to energy efficiency. The main focus is methods that can be applied onboard in operational optimization software, which can aid the crew in operating the ship in a more energy efficient manner, or be integrated in the ship's control systems.

The thesis consists of an overview report and of two research papers and a technical report written during the PhD project. These are added as appendices. A journal and a conference article have been published. We are also in the progress of making an article based on Chapter 4 in this thesis. The conference article is also in the progress of being published in a journal.

Lyngby, October 2011

Jóan Petur Petersen

Papers included in the thesis

[A] Jóan Petur Petersen, Dánjal Jákup Jacobsen, Ole Winther Statistical Modelling for Ship Propulsion Efficiency *J. of Marine Science and Technology*, 2011. Accepted for publication.

[B] Jóan Petur Petersen, Dánjal Jákup Jacobsen, Ole Winther A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions *10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11)*, 2011. Presented and published. Will be reprinted in the *J. of Ship Technology Research*.

[C] Jóan Petur Petersen, Dánjal Jákup Jacobsen, Ole Winther Gaussian Mixture Models for Analyzing Operational Ship Data *Techincal Report*, 2011.

Acknowledgements

I wish to thank my supervisors Ole Winther and Daniel J. Jacobsen, and my colleagues at D3 and DTU Informatics.

I also wish to thank the Open Source community. The PhD relies heavily on Open Source tools such as Python, NumPy, SciPy, and the open source tools on which they are built.

I wish to thank Strandfaraskip Landsins (SSL) who own the ship Smyril where the data was collected. This was the original ship where the first data collection system was installed.

The PhD project is funded by Føroyagrunnurin frá 1971 which is administered by Granskingarráðið.

Contents

Chapter 1

Introduction

The objective of this thesis is to investigate how data driven models can be applied to conserve energy used for propulsion onboard ships. Data driven modelling refers to empirical modelling using machine learning methods. The models are based mainly upon observations, and the prior knowledge about the physical system is limited. A data driven model that describes a ship's energy usage as a function of the ship's state and the surrounding environment can be used to minimize the energy required for propulsion.

One might ask why use data driven models, when there already exist methods for estimating the resistance of ships, e.g. Computational Fluid Dynamics (CFD), first principles models, tank tests, etc. Such models play a vital role in the design of ships, and is still an area of ongoing research. But these method have their drawbacks for the purposes we intend to use them. By using data driven models generated from real data, it is hoped that we get a better representation of the actual ship; capturing effects not described by the other models, ranging from complex interactions with the sea to deviations from the ship design to the ship actual built. The data driven models are not intended to be competitors with already existing approaches, but more as an additional tool.

This thesis examines data driven propulsion models; this involves preprocessing, explorative analysis, and modelling of the data, and evaluation of the final models.

1.1 Project background

The PhD project is a collaboration between a Faroese company Decision3 and the Danish university DTU. The project is funded by the faroese foundation Føroyagrunnurin frá 1971, which is administered by Granskingarráðið.

The PhD project emerged from a research project "Level Measurement System", which was funded by Granskingarráðið. This project aimed at finding a suitable way to measure the distance to the sea surface, and from it derive information about the placement of the hull in the sea, and the state of the sea. It was during this work that we became aware that it would be a good idea to model the ships resistance from collected data, instead of only relying on traditional methods, which in many cases really are intended for the design of the ship, and not for the operational stage.

In a free market or a market economy the price of oil is controlled by supply and demand. The influence that the Faroe Islands can exert upon the oil price is therefore close to none, due to the country's size. But the oil price has a very profound effect on the Faroese economy, as its primary industry, the fishing industry, is highly dependent on oil. So the oil price is a significant factor controlled from the outside that the Faroe Islands have to adjust to in order to have a healthy economy. One possible way of adjusting is to make the production more energy efficient by conserving energy. And work has already been done in this area in the Faroe Islands, for example more efficient trawl doors and trawls have been developed.

Besides the economical advantages of reducing energy consumption, there are also ecological benefits. Using less oil reduces the emission of greenhouse gasses and other pollutants. If the Faroe Islands will enter the Kyoto amendment then knowledge about the energy consumption onboard ships will be valuable information for reaching the mandatory targets given by the amendment. According to a report [20] published by the Faroese Food- veterinary- and environmental agency 32% of the carbon dioxide emissions in the Faroe Islands where from fishing vessels in 1990.

1.2 Online resources

There are two online resources associated with this PhD project; the project homepage and a publicly available pattern and machine learning library.

The project has a homepage located at

http://cogsys.imm.dtu.dk/propulsionmodelling,

where the data and code used for the thesis and articles can be downloaded. Instructions can be found on the homepage for how to run the models. The data is presented in a raw unprocessed form so as not to restrict and leave the decision open for other researchers. For convenience the preprocessed data is also available, which is used to obtain the results presented.

The second resource is a Python library for pattern recognition and machine learning, called PyPR (**Py**thon **P**attern **R**ecognition). It is located at

http://pypr.sourceforge.net.

It contains most of the algorithms used for this project, and accompanying online documentation. The code is released under GNU General Public License, as it is hoped that other people will find it interesting and contribute to its further development.

The rest of this Chapter is organized as follows: In Section 1.3 some of the basics behind ship propulsion. In Section 1.4 the setup used on board the ship where the data is collected for this project is presented. An the finally in Section 1.5 the data preparation process for modelling is described.

1.3 An Introduction to Ship Propulsion

This section is intended for readers not familiar with ship propulsion and the environment ships operate in. The focus will be on the energy efficiency of the process. Marine propulsion is a very complex field covering several scientific disciplines. However this introduction should give the reader the necessary insights to grasp the ideas behind the models developed later in this thesis and the reasoning behind the development of these.

1.3.1 Main Propulsion System

The main propulsion system consists of the main engines, propellers, and propeller shafts. The purpose of the main propulsion system is to convert energy

or fuel into thrust to move the ship across the water. The power is generated in the engine and propagates through the gear-box and shaft to the propeller, where it is converted to thrust. Power is lost through friction, heat, and other factors. The efficiency of the engine will depend on several factors, and will change over time. For example, engine temperature plays a role in its efficiency. The oil used also plays an important role in the efficiency of the engine.

Most propeller systems can be categorized as *fixed* or *variable* pitch. In a fixed pitch propeller the angle of the propeller blades can not be changed. Various amounts of thrust is generate by rotating the propeller at different rotational speeds. A variable pitch propeller has blades where the angle can be changed so as to control the thrust generated, so here we have one more dimension of freedom. However, in practice, the rotational speed is often kept constant, for example due to the presence of a shaft-driven electricity generator, which may also be connected to the shaft. So there is only one operational control left, the propeller pitch. [23] The ship requires a considerable amount of energy for electricity for the onboard equipment, such as the pumps, lighting, compressors, cooling, etc. There are two common ways of generating the electricity, either using a shaft-driven generator or a generator running on an auxiliary engine. A shaft-driven generator will therefore act as an additional load on the main engine, and will have to be taken into account when looking at the propulsion efficiency when it is based on the main engine fuel consumption.

In addition to the main propulsion system, the ship also has thrusters built into the hull for transversal propulsion. These are typically used for maneuvering. We will not look at these further.

1.3.2 Rudders

The purpose of the rudders is the control the direction of the vessel. The rudders is normally placed just behind the propellers. When a rudder is turned forces will act upon it and the hull, making the ship turn. It will also create additional drag, and make the ship slow down affecting the propulsion efficiency. When cruising the rudders are use to hold the course. This can be done by a crew member or an auto-pilot.

1.3.3 The Hull

The hull is the body of the ship itself. The waterline of the hull is the line to which a vessel's hull is immersed in the sea. Above the hull there is normally

some kind of superstructure, often a deckhouse or similar. When moving the vessel the part of the hull submerged below the waterline encounters resistance due to the water, and the part of the ship above the waterline will meet resistance from the air. The *draft* is the distance from the waterline to the keel amidships. This relates directly to how heavily the ship is loaded. The *trim* is the difference between the ship's draft aft and forward. The trim of a sea-going vessel can significantly affect fuel efficiency ([17],[8]). One of the reasons for this is that the trim affects the angle and position of the bulbous bow of the vessel, but in general it is no surprise that changing the pitching angle of the vessel in the sea has a significant effect on propulsion resistance and thereby efficiency. Even for vessels with smaller bulbous bows, the position and angle of propellers, rudder and the shape and size of the wetted surface is affected by trimming.

The shape of the hull plays a vital role in the resistance. When the ship is pushed through the sea by the propeller, it will generate resistance against the force. A common criterion in the ship design process is to minimize this resistance. The resistance will not only depend on the shape of the hull, but also on its placement in the sea and its dynamics, which can change due to loading of the vessel and so on.

The surface of the hull also plays a vital role in the resistance. The resistance will depend on the type of paint used, and on the amount of fouling on the hull; organic material that will accumulate on the surface over time. Most vessels have antifouling or so-called 'fouling release' paint covering the underwater hull. The propeller blades are generally polished metal, and will normally have no antifouling provision [40]. Three common types of hull fouling are slime, shells, and weeds. These are also referred to as micro- and macro-fouling. Slime being microfouling, consisting of unicellular algae and bacteria. Shells and weed being macrofouling. Full scale studies have shown that an 8-18% decrease in total required propulsive power can be attributed to the removal of slime film [40]. It is also difficult to predict how the slime will develop on the hull for a particular ship, route, or coating.

A well known example is the classic pontoon tests of Kempf (1937) [40]. He found that the maximum drag increase occurred with shell fouling covering 75% of the wetted surface. The shell height was 14 mm. But even with only 5% of the wetted area covered, the drag increase was 66% of the maximum. From his empirical data he was able to calculate the increased resistance of ships due to shell, in a number of cases. For example, a 120 m vessel, with 75% coverage with shell of 4.5 mm height would show an increase of 85% in skin friction resistance.

It is evident from these numbers that fouling contributes considerably to the fuel consumption required for propulsion. The ship owner must consider the costs of cleaning the ship, which mostly means that it has to been taken out of

service. A valuable tool in the decision making would be a tool, which could estimate the potential savings of cleaning the hull or propeller blades.

1.4 Data Collection System Setup

In this section the data collection system used to collect the publicly available data set will be presented. The important devices and measurements will be described here.

1.4.1 Data collection system

The data collection system plays a vital role in a data driven system and must therefore be reliable and stable. This section gives an overview of the data collection system, and some details about the important signals that it collects.

The data that we will use for generating the efficiency models is collected on board a ferry, Smyril, which services a domestic route in the Faroe Islands. The ship serves a daily route from the capital Tórshavn to Tvøroyri on the south most island Suðuroy. The ferry makes two to tree trips back and forth each day, and the duration of each trip is around 1 hour and 55 minutes. The ship is designed by Knud E. Hansen, and built on the IZAR shipyard, San Fernando, Spain, and was delivered in 2005. According to Knud E. Hansen, the vessel was especially designed for this route which is classified as open sea in the North Atlantic.

A computer is placed onboard the ship, which collects data from several selected devices. The data from the devices is sanity checked, down sampled, and stored in a database for further analysis. Each data collection point has some cost: cabling, connections, software, storage, etc. So while it is important to measure all relevant signals, there is also a limiting cost and complexity factor to consider in each case.

The system collects data from the following devices: The Doppler speed log, gyrocompass, GPS, main fuel pipe flow meter, rudder angles, wind, and propeller pitches. In addition to these an inclinometer has been installed in the engine room to measure the trim of the vessel, and two level measurement devices have been placed on the sides of the ship to measure the distance down to the sea surface. In future setups like this it should be considered also using accelerometers, as these are cheap and do not require much setup while provid-

ing meaningful data on ship motion, which affects fuel efficiency. In the case of Smyril, the water depth below the keel and the state of the stabilizers could also be of interest, but are not measured. Including these would probably improve the precision of the model considerably in certain conditions.

1.4.2 Wave and draft measurements - Level Measurements

The system has two level measurement devices installed on the sides of the ship. One on the bridge wing on the port side, and one midships on the starboard side. These use radio waves that measure the distance down to the sea surface.

Figure 1.1 shows how the starboard radar is mounted. On Smyril the port radar is positioned 19.3 meters above the bottom of the hull, and it points 19 degrees away from the hull. It is placed -9.0 meter from the center of the ship (negative to indicate it is behind midship). The starboard radar is placed on the same level as the bridge. It is located 22.1 meters over the bottom of the hull (keel), and it has an angle of 12.6 degree forward - it does not point out from the hull as the other radar. It is placed 29.8 meters forward from the middle of the ship.

Measurements which are too close to the radar, likely because the radar has locked onto something on the hull, and not the sea, are removed by the program reading the data from the radar. Each radar can sample at up to 60 Hz if it locks onto a good reflective surface. This rate is much higher than the sample rate, 3 samples/second, that is considered sufficient for modelling purposes. To avoid too much data being stored from the radars, the signal is passed through a median filter. Currently a median window of $\frac{1}{3}$ seconds is used.

1.4.3 Trim - inclinometer

The inclinometer is installed in the engine control room. We have calibrated it by measuring the angle and comparing it to the draft marks on the hull. We found that 357.34° as measured by the inclinometer corresponds to 0.00° on the vessel. The inclinometer used is a Seika.de NG360. It is a capacitative liquid based inclinometer. According to its specifications it has a resolution of 0.01 degrees, and a settling time of approximately 0.3 seconds. It seems, however, that there is a problem with the data from this unit - the measurements seem to lock occasionally. The source of this error is not found yet, but it could be due to mechanical vibrations in the engine control room.

Figure 1.1: Radar mounted on starboard side.

1.4.4 Heavy fuel oil flow meter

Figure 1.2 shows a simplified version of the fuel piping onboard Smyril. Smyril can also use diesel fuel, but these pipes have been omitted. The HFO (heavy fuel oil) service tanks hold fuel oil used by the engines; they are drawn as a single tank in the figure. These pipes go to the so-called booster modules, which have internal mixing tanks. According to the crew, the mixing tanks can hold around 60 L. Each booster module supplies two engines, and receives the return fuel from the same engines. If there is excess fuel in the mixing tanks, then it is returned to the service tanks through the return pipes.

Initially an Eesiflo 7000 flowmeter was used to measure the main engine supply and return flows. This flowmeter works by using ultrasound, and is clamped onto the oil pipe. EESIFLO uses ultrasonic signals for the measurement of liquid flow, employing the transit time to measure the flow. Ultrasonic signals are emitted alternatively between to transducers in the direction of the flow and against it. Because the medium in which the signals propagate is flowing, the transit time of the sound signals propagating in the direction of flow is shorter than the transit time of the signal propagating against the direction of flow. This solution was selected because it is a non-invasive method; the transducers are just clamped

onto the pipe, it permits safe measurement on aggressive or high temperature media flowing in closed conduits. Another advantage of this solution is that the flow can be measured without interruption of Smyril operation, and that the installation did not require any alterations to the pipe system. According to the Eesiflo 7000 manual, the precision is 2-3% of read value.

The ultrasonic flow meters were installed on the supply and return pipes from the supply tanks, as shown in figure 1.2. There is very rarely any non-zero measurement in the return pipe from the two booster modules. This is also consistent with what we have been told by the crew. Maybe the return pipe is only for flushing or for irregular overflows. Therefore, it might actually be acceptable to use just one flow meter on the supply pipe.

There are some drawbacks to the current setup. In the booster modules there are mixing tanks, in order to incorporate the return flow from the engines to the supply line flow. The fuel level in this mixing tank can change, and this will have an effect on the momentary fuel consumption recorded by the flowmeter - a rise in the tank level will seem like a higher fuel consumption - if the level is falling, the recorded fuel consumption will be too low.

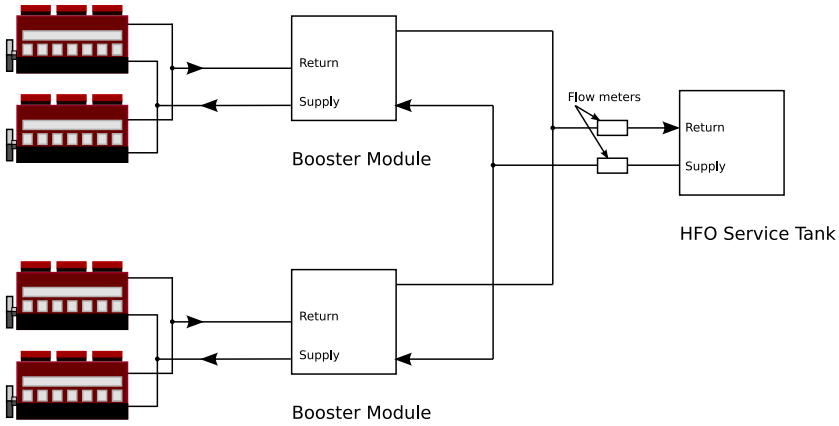

Figure 1.2: Simplified fuel piping onboard Smyril.

There are two other options for installing flow meters in this piping system. One would be to install four flow meters on the supply and return pipes from the booster modules for the engines. And the most radical would be to install flow meters on the supply and return on each engine, totaling in eight flow meters. The advantage of installing four flow meters would be that mixing tanks in the booster module would not affect the measurements. With eight flow meters small changes in system would be noticeable in the fuel flow measurements quickly and in more detail. However such a setup is also four times as expensive

than measuring all the engines as with the current setup.

Unfortunately, it soon became evident that the measurements obtained with the ultrasonic measurement devices where not good enough. The flow measurement seem to be quite good at times, but often the flow meter seemed to lock unto some wrong signal, so that the recorded flow was scaled by some factor. There may have been several reasons for this - maybe the installation was not performed correctly, as the installation crew was not very experienced working with such equipment. The PhD project was considerably hindered by this problem, because the data itself plays such a central role. It was discovered that the there was systematic errors in the measurements; it seemed like the ultrasonic flowmeter switched between a set of discrete factors. In an effort to save the data we tried to model this behavior with a Hidden Markov Model (HMM), but this was mostly unsuccessful. After much hassle it was finally decided to buy another flowmeter, which now seems like a wise decision. But it also took time to get this flowmeter working and collect data, so much time was wasted on faulty measurements. This story only goes to show the nature of the problems, which such a system faces, and the importance of reliable measurements for the critical parameters. The flow meter was replaced with Micro Motion F100 FlowMeter, which resulted in consistent fuel measurements. The new meter exploits the Coriolis effect to measure the volume flow and density.

Looking back it would have been a good idea to do more measurements on the engines, for example we could have measured if some engines where running on diesel. The motor temperature would also be of interest, and the RPM. We did not select to measure the RPM due to a very limited budget, and because it is in principle constant. However looking back these might not have been optimal choices.

1.4.5 Similar systems

The introduction given in the article in Appendix A gives an overview of similar systems.

In the conference article “Development of an On-board Performance and Trial Trip Analysis Tool” by Lennart Pundt [34] a tool is presented for on-board performance and trial analysis. Here the focus is on the trial trip, and the system is able find the correlation between engine power and ship speed. This system has a user interface which assists in the trials, and can therefore be used to access the quality of the results immediately. Finally the system performs an offline analysis of the collected data, and reports are generated for the yard, owner and classification society.

1.5 Data Preparation

The data is stored in a database onboard the ship. Before the raw data can be used to generate models, the data needs to be prepared and cleaned. This sections summarizes some of the most important aspects of this process.

1.5.1 Pre-processing

As most of the equipment on the ship pushes or broadcasts the values to the data collection system, there is very little control with the sampling times. Building a framework to handle this asynchronous data allows our model to be more flexible.

The asynchronously sampled data can be described as a set of 3-tuples containing N_R samples

$$\Gamma = \{(i_1, t_1, v_1), (i_2, t_2, v_2), (i_n, t_n, v_n), \ldots, (i_{N_R}, t_{N_R}, v_{N_R})\} \tag{1.1}$$

where i_n identifies the input, t_n is the timestamp, and v_n the recorded value. The timestamp gives the time at which the value was recorded.

A window can be defined as a period in time, so that $t_1 \leq t_n < t_2$. The elements within that window are then given by a sample window

$$W_{t_1,t_2,j} = \{(i_n, t_n, v_n) \,|t_1 \leq t_n < t_2, i_n = j\} \tag{1.2}$$

where W is the set of samples within that window, and j is used to select a certain signal. A feature extractor is a function, which takes a sample window, and returns one or more feature values based upon the window's contents. As we can see, there are already many degrees of freedom.

Missing values and outliers are also a problem. The system collects values from many sources, and one of them might fail. The current solution to missing values, is to require at least one sample of the required inputs be present in each window, otherwise the window will be dropped. Outliers are found by specifying an acceptable range for the input. If a value is outside the range, then it will be dropped.

For simplicity window size has been fixed to 3 minutes for all the regression models given in Chapter 2, and 15 seconds for the dynamical models examined in Chapter 4. However it is likely that a better performance can be obtained by investigating this further, for example by having an individual and possibly different window size for each individual feature.

One of the inputs, the wind direction, has a circular discontinuity. The wind angle is defined as a number in the range $[0; 360]$ degrees. So when the wind angle passes between 0 and 360 degrees, the mathematical model will experience a discontinuity; even though the two adjacent values are closely related. One solution would be to divide the input into categories. For example we could have one category for $[0; 45]$, one for $[45; 90]$, and etc. But there are drawbacks to this approach. The precision will depend on the number of categories, and if we increase the number of categories, we will introduce many new inputs. Another approach will therefore be used, where we project the measurements into two new dimensions called headwind and crosswind. From the two inputs signals, relative wind speed and angle, we create two new signals according to equation 1.3 and 1.4,

$$V_{hw} = V_{wind} \cdot cos(\theta_{wind}) \ , \tag{1.3}$$

$$V_{cw} = V_{wind} \cdot sin(\theta_{wind}) \ , \tag{1.4}$$

where V_{hw} and V_{cw} gives the equivalent headwind and crosswind respectively, given the wind speed V_{wind} and wind direction θ_{wind}.

Some basic corrections to the radar signal are done in the preprocessing step; corrections are done for the angle of the radar so that the distance is transformed to be the distance vertically down to the sea surface, Eq. 1.5.

$$f = cos(a/180.0 * \pi) \tag{1.5}$$

$$y = h - f * x \tag{1.6}$$

Where, x, is the raw radar signal in meters, h, the radar's height above keel, and, a, the radar's angle to vertical.

More work could be done analyzing the data, for example it could be interesting to extract the significant wave height and direction, and do some power spectrum

analysis of the wave lengths and there examine the relationship between this spectrum and the propulsion efficiency.

1.5.2 Feature extraction

The purpose of the feature extraction step is to extract the relevant information for the problem from the sample window. This is a delicate process, because we reduce the information contained in several samples, into one or a few feature values. The feature extractors we have selected to use are: Mean, variance, and the derivative. See Appendix A and B for the definition of these. Many other feature extractors could be used, for example min, max, entropy, filtering, and so on; the efficiency of these depends very much on the application. When using regression for prediction within one window, one should take care of not using any features, which use the target output, as this is considered unknown.

1.6 Thesis outline

The thesis is organized as follows. Chapter 2 gives an overview of the regression model mainly focusing on the two articles given in Appendix A and B. Chapter 3 introduces the dynamical modelling approach using the well known sunspot time series. Chapter 4 extends the modelling approach presented in Chapter 3 to work with ship time series. Chapter 5 contains the thesis conclusion.

Chapter 2

Regression Models

Regression is one of the more straight forward approaches to ship propulsion efficiency modelling - it has been studied in the literature before, and has many useful applications. The first articles published dealing with marine propulsion efficiency, approached the problem using instantaneous regression models [21, 28, 27, 14, 13]. The two first articles published as part of this PhD project also examined regression modelling. These articles are given in Appendix A and B. The technical report included in Appendix C also has some results based on the regression modelling problem.

The regression model will not be examined in great detail in this chapter, as there are already two articles that look at this model approach in detail attached as appendices. Instead a short overview of this modelling approach will be given, and some ideas not mentioned in the articles will be presented.

The rest of this chapter is organized as follows. Section 2.1 gives an overview of the regression modelling approach. Section 2.2 introduces feature ranking using the Gaussian mixture models. Section 2.3 examines two applications for the regression model, a trim optimization application and an application for estimating the effect hull fouling has on the propulsion efficiency.

2.1 Modelling Approach

The regression model relates a set of feature values to an output at a certain point in time, and may be considered static compared to the dynamical models that will be investigated in Chapter 4.

From the regression model's point of view the controls and states are not different kind of features, and will therefore not be handled separately in this Chapter. The vector $\mathbf{x}_n$ will be used to represent the inputs to the model. The vector consists of the union of the controls and various state measurement features. The index n is used to identify the window number from which the features have been derived. The preprocessed data will consist of $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)|n = 1, \ldots, N\}$ of N input-output pairs. The estimated or target variables are called $\mathbf{y}_n$ and could for example be the speed of the vessel, fuel consumption, or the fuel efficiency. The modelling problem can therefore simply be described as a conditional distribution $p(\mathbf{y}_n|\mathbf{x}_n)$: given some ship state and control settings, what is the probability of a given target.

The Artificial Neural Network and Gaussian Process are based upon the conditional probability distribution $p(\mathbf{y}_n|\mathbf{x}_n)$, and are described in detail in the first paper "Statistical Modelling for Ship Propulsion Efficiency" given in Appendix A. In the technical report in Appendix C an alternative approach using the joint distribution, $p(\mathbf{x}_n, \mathbf{y}_n)$, represented with a Gaussian mixture models is described. The conditional distribution can be obtained from the joint distribution, which can be obtained cheaply from the Gaussian mixture model. The conditional distribution will also be a mixture of Gaussians. Models that describe the conditional distribution are often referred to as *discriminative*, while models describing the joint probability are called *generative*. This is discussed in detail in the technical report in Appendix C.

The regression model for predicting the fuel consumption and speed from a set of measured features using a discriminative model [3]

$$\mathbf{y}_n = f(\mathbf{x}_n; \theta) + \epsilon_n \ , \tag{2.1}$$

where f denotes the model function depending upon parameters θ and explanatory (or input) variables $\mathbf{x}_n$. The residual ϵ_n is the part of the measured signal $\mathbf{y}_n$ not explained by the model due to noisy measurements and/or model shortcomings. The model's parameters θ are learned (inferred) from the training data.

The Artificial Neural Network, the Gaussian Process, and the conditional distribution derived from a GMM can all be used to represent the function f in Eq. 2.1. All models can successfully model the speed and fuel of the vessel.

However all three have their advantages and disadvantages, which are discussed in the articles. The Gaussian Process might at first seem like the ideal model to use for the ship efficiency modelling, because it provides an uncertainty measure with the estimates. However the GP is computationally demanding, which can be problematic if use in an onboard online system, where the time used to obtain the results plays an important role. The computational demands of the GP was one of the problems that lead to development of the technical report in Appendix C, where Gaussian Mixture Models are used to achieve similar results within a more reasonable amount of time. One of the Gaussian Process strengths and weaknesses is its non-parametric nature. The non-parametric nature causes problems with large data sets - the complexity of the model grows with the size of the data to the third power. Work has been done improving the scalability of GP, see Ref. [38] or [37]. Another problem with GP is that they do not handle data of high dimensionality very well. It therefore does not seem that the Gaussian Process is a good choice for an onboard system, if these problems are not solved in some manner.

The data or model code used in articles dealing with ship propulsion modelling [21, 28, 27] is not publicly available, so comparing results is of little value. This does not seem to be a tradition within this field. It is unfortunate, as it is impossible to reproduce the results or compare them. If the model code was made public, the it could be tried on the public dataset given in this thesis, and the modelling approaches could be compared.

2.2 Feature Ranking

One of the interesting features with Gaussian Processes, is that it has a built in feature relevance measure, see the article in Appendix A. This can be very useful for ranking the importance of each feature, and also to find out how much influence the actual measurements have on the final result.

The Gaussian mixture models presented in the technical report in Appendix C can to a certain extent be used to find the importance of the features. This can be done by removing one or more features, and the evaluate the performance of the model again. The removal of features is cheap in a Gaussian mixture model, as the given features can be marginalized, and no re-training of the model is needed. Figure 2.1 gives a plot where pairs of features have been removed from the regression data set. As expected the most important features for predicting the speed of the vessel are the propeller pitches. The Gaussian mixture model is able to obtain good results for the speed of the vessel when only one of the propeller pitches is removed - this because they propeller pitches have very

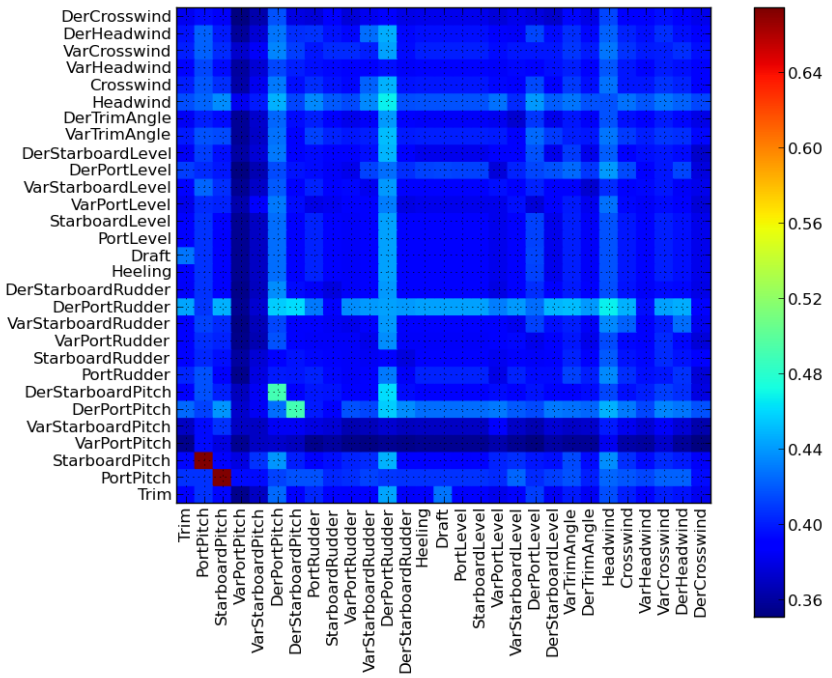

Figure 2.1: The performance obtained by removing pairs of feature inputs.

similar values most of the time. It is only when maneuvering in the harbor they might diverge for some time. The information provided by the starboard and port propeller pitch features are mostly the same most of the time. After the propeller pitch features, the derivate of the same signals are the ones who come as the third and fourth most important for the Gaussian mixture model.

The data used is the test data from the data split up by trips, see Appendix A for an explanation of the regression data sets.

2.3 Applications

The article "Statistical Modelling for Ship Propulsion Efficiency" given in Appendix A gives many example of areas where statistical models can be applied for optimizing the ship propulsion efficiency and operations.

Here two applications will be examined closer in relation to the regression model. First a trim optimization application is described, and then a fouling penalty estimator.

2.3.1 Trim Optimization

The idea behind the trim optimization system, is to find the optimal trim for the ship in the given conditions. The regression model could be used for this. Having the fuel efficiency described by the model, the control space can be searched for the optimal set of controls. However there is a problem. The regression model knows how to map the inputs to a few target outputs - it does not know how the state of the vessel is affected by the control change.

A trim optimization system only based upon a regression model will have a behavior, which can under certain circumstances undermine the crew's confidence in the system. The system will be based upon an assumption that changing the trim or speed setting will not affect other state variables, such as the squat, autopilot behavior, dynamics, and etc. This will be experience by the crew the following way: The system will give advice on an optimal setting. The crew will then start working on complying with the given advice, but in the meanwhile the assumption that the state variables will not change may be false, and another optimal point will be calculated by the trim optimization system as the conditions change, and the advice given to the crew will suddenly change.

Such an approach is not viable, and another dynamical model approach will be given in Chapter 3 and 4 which solves this problem.

2.3.2 Fouling Model

In Chapter 1 it was explained how fouling will have an effect on the propulsion efficiency of the ship. This fouling penalty will grow slowly over time. In an article by B. P. Pedersen et al. [28] it is suggested that the regression model can be used to predict this penalty. Due to the short duration of our data set, it has not been possible to detect any slow temporal decline of the performance, which one might expect from fouling.

As simple approach to estimate the fouling penalty, would be to have a baseline model trained for a duration where the ship hull can be considered clean. This model would then be used to estimate the performance of the ship over time, and the difference between the estimated and the measured efficiency would

tell something about the state of the hull and propeller fouling. The difference between the estimated and measured efficiency would of course have to be processed in some way first, as it will be very noisy. Another approach for including the fouling penalty, would be to have a feature input to the model, which counts the time since the last hull cleaning. A similar feature could also be included for the time since last propeller cleaning. This way there will be no need for any baseline model, and the whole data set can be used.

2.4 Conclusion

The problems associated with using the regression model for a trim optimization system is a good motivation for developing the models further. In the next two chapters dynamic propulsion models will be developed. These models take the temporal sequence of the data into account, and can solve some of the problems associated with the regression model.

Feature ranking is mostly interesting from a modelling and system analytical point of view. In practice it might help make the data collection system cheaper, as one might find that some of the measurements are unnecessary for the applications at hand, and can therefore be omitted in the next installation of the system.

Further work should also be done with the regression model approach. The regression model might be better suited for some applications than a dynamical model. One approach which might be interesting to explore, is to combine the regression model with model based upon first principle models. This could reduce the training time needed for the model, and make it better at extrapolating into unexplored areas.

Regarding the fouling penalty, a data set should also be obtained where the fouling penalty can be estimated from - this data set should preferably contain one or more hull cleaning events, so that these events can be studied closer.

Chapter 3

Dynamical Modelling

The regression models presented in Chapter 2 do not take the sequential nature of the data into account - they predict the instantaneous target variables from a snapshot of the state of the ship. The temporal and dynamic information is therefore not reflected in the regression models. A regression model will not be able to determine how control changes will affect the variables describing the state of the ship. In this chapter a *dynamical modelling* approach is presented, which can solve these problems to a certain degree.

The article presented at COMPIT'11 given in Appendix B introduces the idea of modelling the propulsion parameters of the ship using time series forecasting. This paper gives a picture of what direction we where working in, at the time. This is likely the first paper, which examines this kind dynamical modelling for ship propulsion.

This chapter is organized as follows. Firstly the time series and forecasting are defined. Secondly the well known sunspot data set is used to introduce the basic concepts of time series forecasting, so as to introduce the problems encountered in the more the complex ship data set. This chapter is intended to be an introduction to chapter 4, where the dynamical model for the ship propulsion problem is described and examined.

3.1 Time Series Forecasting

We define a time series, $\mathcal{X}$, as a sequence of N vectors X_n

$$\mathcal{X} = \{\mathbf{X}_n | n = 1, 2, \cdots, N\} \ , \tag{3.1}$$

drawn from a stochastic process. The index n is used to identify the sample vectors. Here only uniform temporal sampling will be used, so that the indicies can be mapped into time as given in Eq. 3.2,

$$t = t_1 + T(n-1) \ , \tag{3.2}$$

where t_1 is the starting point in time for the sequence, which corresponds to the sample $\mathbf{X}_1$. The *sampling period* is given by T and is the duration between the samples.

Figure 3.1 illustrates a simple model setup, which takes part of time-series sequence of vectors as input, and maps it into a prediction of the next step vector. The model takes d steps, $\mathbf{X}_{n-d+1}, \ldots, \mathbf{X}_n$, of the times series as inputs, and returns a vector of the predicted next vector $\tilde{\mathbf{X}}_{n+1}$. The boxes with a Δ work as time delays or lags, so that the network can store previous values - if the input to a lag is $\mathbf{X}_n$ the output will be $\mathbf{X}_{n-1}$. The error $\mathbf{e}_{n+1}$ is assumed additive, and is explicitly drawn after the model. The dimensionality of the sample space presented to the model f grows linearly with the number of time steps.

The time-delay model can be described as given in Eq. 3.3,

$$\tilde{\mathbf{X}}_{n+1} = f\left(\mathbf{X}_n, \mathbf{X}_{n-1}, \ldots, \mathbf{X}_{n-d+1}, \mathbf{w}\right) + \mathbf{e}_{n+1}(\mathbf{w}) \ , \tag{3.3}$$

where f is a function or model that maps the seen sequence of sample vectors into a predicted future sample, w are the model's parameters often also called weights, and e is the error or residual for the prediction.

We will distinguish between *measured* $\tilde{X}_n$ and *predicted* $\hat{X}_n$ variables. Variables presented to the model will be called X_n and might either be measured or predicted depending on the situation.

In order for the time-delay network to track a time-series, it has to be started of in some position with measured values. If doing forecasting several steps

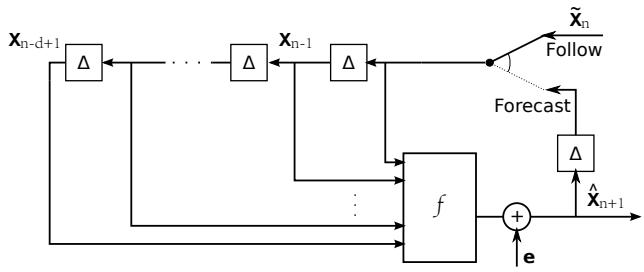

Figure 3.1: A time-delay network. The heart of the system is a model which maps a sequence of historic state vectors in to a new predicted state vector, $\tilde{\mathbf{X}}_{n+1}$. This model is illustrated by the rectangle with an f inside of it. The measured signals enter in the upper right corner of the diagram. Here a switch can let them through, and they are passed onto the model as inputs. The rectangles with a Δ inside of them are time delays, meaning that for each tick in the system the output of the time delay is set to its input. The position of the switch can be changed, so as to let the predicted next states be used instead of the measured states.

into the future, the network will have to feed the results to the inputs. This is illustrated by the switch in Figure 3.1, where the time-delay network follows the measurements when the switch is in the up position, and forecasts in the down position.

3.2 The Sunspot Time Series

As to introduce time series forecasting, we will start by looking the sunspot data set. The sunspot examples that are presented here exemplifies many of the pitfalls of time series modelling, and are not as complex to understand as the dynamical ship models, where many more factors play a role.

The data that will be used gives the average yearly sunspot activity, starting from 1700 and continues to 1979 including both. The sampling period for this data is therefore $T = 1$ year, and $t_1 = 1700$. The samples have been scaled, so that the highest average sunspot activity is 1.0, which corresponds to an average sunspot number of 179.1. The sunspot data set can be obtained online from NOAA National Geophysical Data Center [25]. However the data and examples here have been included in the PyPR package [29].

Figure 3.2 gives the power spectrum of the sunspot number, and the histogram

of the occurrences. The number of sunspots depends on the solar activity, which has a dominant periodicity at about 11 years.

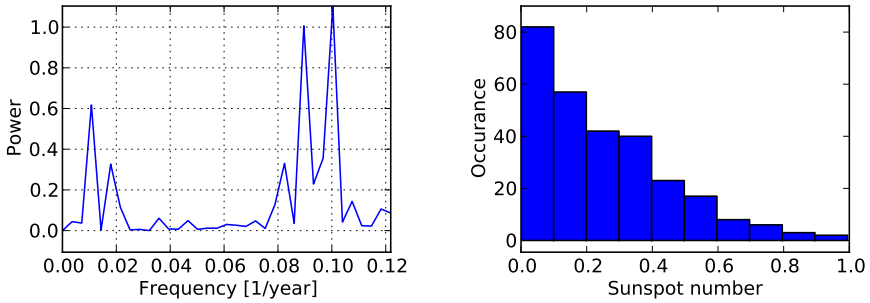

Figure 3.2: The frequency power spectrum and histogram of the sunspot numbers. The dominant period can be seen in the power spectrum. Another period has almost the same amplitude.

Using the samples from the whole sunspot time series, we can find the highest of the 5% lowest values, and the lowest for the 5% highest sunspot values - then 90% of the sunspot values would lie in between these two values. It will be called the 90% percentile interval. This is done in the same way for 50% percent quantile. Table 3.1 gives some statistics on the sunspot dataset.

Short alias	Description	Value
$x^{(5\%)}$	$\{x \mid P(X \leq x) > 0.05\}$	0.02
$x^{(25\%)}$	$\{x \mid P(X \leq x) > 0.25\}$	0.08
$x^{(50\%)}$	$\mathrm{P}(X \leq x) \geq \frac{1}{2}$ and $\mathrm{P}(X \geq x) \geq \frac{1}{2}$	0.20
	$E(X)$	0.25
$x^{(75\%)}$	$\{x \mid P(X \geq x) > 0.75\}$	0.36
$x^{(95\%)}$	$\{x \mid P(X \geq x) > 0.95\}$	0.66

Table 3.1: Some statistics on the sunspot values.

As an experiment, the sunspot activity will be predicted based only on historical observations. We will use the data up to and including 1920 as the training data. The rest of the data, from 1921 to 1979 including both is used for testing the final result.

3.3 TDNN for the Sunspot Problem

Using the network given in Figure 3.1 with an Artificial Neural Network as the model is typically called a *Time-Delay Neural Network*, and sometimes a *Tapped-Delay Neural Network*. Both can be abbreviated as TDNN.

The TDNN has been trained using 2/3 of this training data, where the last 1/3 training data has been used for a validation set. The data will be arranged in sections of $d+1$ length. The first part of the section will consist of inputs, and the last sample will be the target. The sections will be randomly distributed between the two sets, and no section occurs in both sets. The validation set is used for finding the weight decay used in the ANN, and to make an estimate of the residual error. The size of the training, testing, and validation sets are somewhat randomly selected, but should be sufficient for these demos purposes. A more proper solution would be to use setup with cross validation, as to find the best regularization parameters.

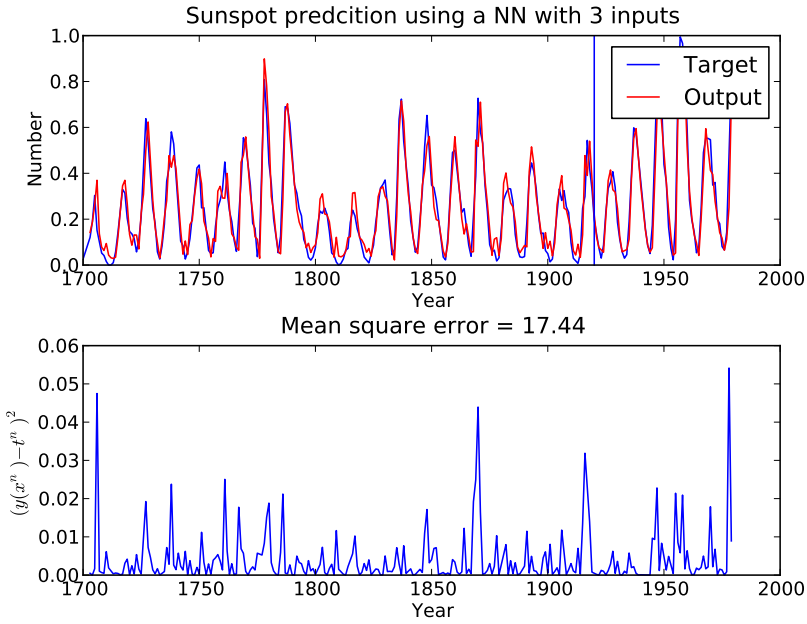

Figure 3.3: A TDNN used to predict the sunspot activity based on historical data. The network looks at $d = 3$ previous time steps and predict the next one. The network here always has measured data as inputs.

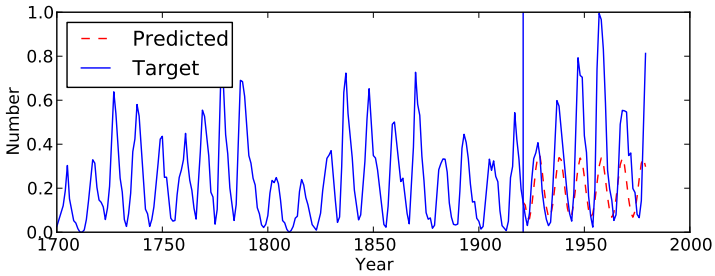

Figure 3.4: A TDNN used to forecast the sunspot activity based on historical data using $d = 3$ lagged inputs. Here the output from the TDNN is fed back to the input.

Figure 3.3 shows the results obtained by using an TDNN to predict one step for the sunspot activity, which means that all the inputs to the model are measured values. The switch given in Figure 3.1 is in the up position all the time. The upper plot shows the correct target sunspot activity, and the predicted output from the model. The vertical line in the plot show the transition between the training and test data. The lower plot gives the squared error. The model seems to perform well predicting the next value for the sunspot activity, given that it is presented with a sequence of measured values. However the results are quite different if the model is used to forecast the sunspot activity further into the future, see Figure 3.4. Here the model is initialized with measured data, and then the switch given in Figure 3.1 is switched, so that the output from the model is feed to the input. The output from the model seems to oscillate at the right frequency, but the amplitude of the signal seems to be quite constant, the model does not capture the variation in amplitude. One of the things that becomes evident from Figure 3.4 is that there is need for some accompanying measure of the error. There will be errors in the model, which will cause it to diverge from the real system, especially over time. Having a forecast without any information about its precision is of little value to us.

One approach to estimate propagated error of the forecasting is to fit the residual error obtained with a known distribution, and use this to sample or generate an random error for each time step. Several particles can be used to simulate the probability density of the propagated forecast error. The particle approach works by having a set of M forecast sequences instances instead of one. Each sequence instance is presented to the model, which predicts the next sample for each sequence, $\hat{X}_n^{(m)}$, where the index m is used to identify the sequence to which the prediction belongs. The residual distribution is used to generate a random sample, which is added to the predicted value $\hat{X}_n^{(m)}$. At some point

n the median, mean, percentile interval, or some other characteristic can be found by looking at the values of the M particles at this index, n, in time. This particle approach has no problems with the model being nonlinear. The 50 % percentile interval, 90 % percentile interval, and median will be used in the plots in this chapter and the next.

The approach used in the COMPIT'11 article in Appendix B, was to fit a Gaussian and a Student's t distribution to the residuals. In the article it is found that the Student's t distribution is a good representive for the distribution of the residuals for the dynamical ship system. For simplicity the Gaussian approach will be used for this sunspot example. We find the residual distribution from error found in Figure 3.3 for the samples selected for the validation, and fit a Guassian distribution to it by giving it the same mean and variance.

Figure 3.5 shows the resulting forecast using 1000 particles to simulate the effects of the accumulated residual noise. The marked area shows where 50% and 90% of the particles are located within. The true sunspot activity lies reasonably well inside the marked areas. It can be seen that the sunspot activity in 1957 is especially high - about 25% higher than any seen before in the training set. This extreme goes outside the 90% percentile interval, as the network and residual representation have no seen such an extreme before, and can therefore not be expected to anticipate this.

The noise will dominate over time. The noise for the $d = 1$ is immediately high, it is a little better for the $d = 3$, where the intervals are quite narrow in the few first 5 years. All of the forecasts seems to flatten out, as the starting condition is lost due to the noise. The $d = 1$ and $d = 3$ seem to flatten out to a similar forecast level and the noise intervals also seem similar - the 50% and 90% percentile interval seem to correspond well with the sunspot characteristics given in Table 3.1. The $d = 11$ forecast seems to settle at a higher level with narrower noise intervals. The percentile intervals for $d = 11$ are narrow and follow the target sunspot well. The noise intervals are bounded within a reasonable area. This is due to the tanh activation functions used in the artificial neural network. The particles will be kept within a reasonable interval even though the noise is added to the state vector at each iteration.

A special case in Figure 3.5 is when $d = 1$ is used. This means that the model only sees one previous value, and will therefore not have any idea if the sunspot activity is rising or falling. It is also reflected in the result, as the model ends up predicting something that looks like a mean value for the sunspot activity. This is a order $d = 1$ Markov model, and even though it seems to perform badly for the sunspot activity, it has some applications in modelling the ship dynamics. The order $d = 1$ Markov model is described closer in Sec. 4.1, and arguments for why it works for the ship data will be given.

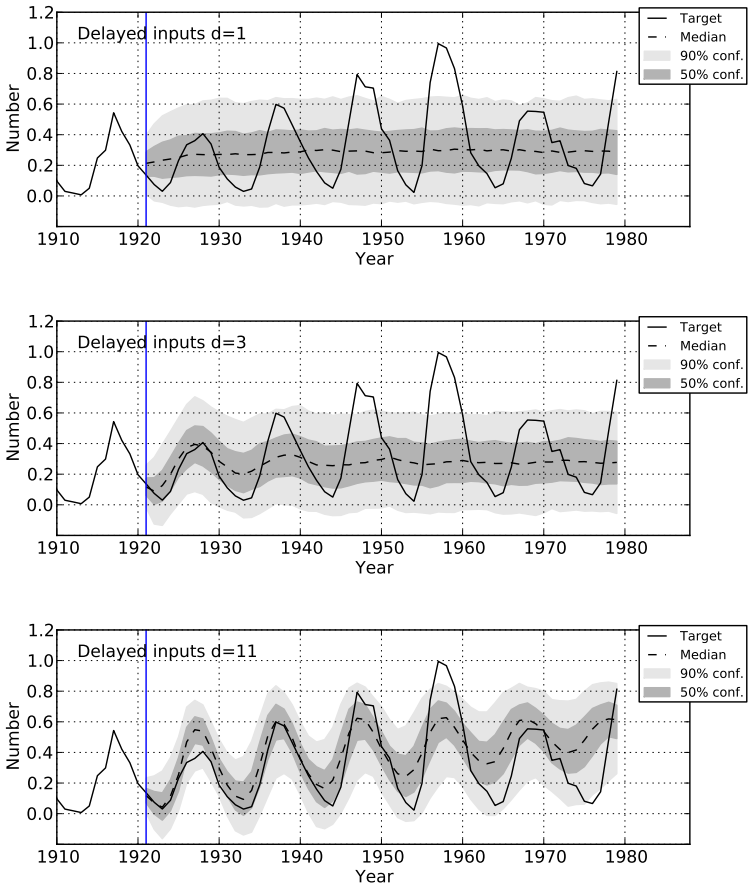

Figure 3.5: TDNN sunspot forecasting. A set of 1000 particles are used to simulate the accumulated residual noise given by the percentile intervals.

3.4 Time Delay GMM Model for the Sunspot Problem

Instead of modelling the residuals of the model, another approach would be to use a model, which could model the conditional probability distribution of the next step directly, see Eq. 3.4.

$$\mathbf{x}_{n+1} \sim p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-d+1}) \tag{3.4}$$

The Gaussian Mixture Model (GMM) will be used to model this probability density function. A thorough walkthrough of the principles using the GMM for conditional distributions is given in the technical report in Appendix C.

Having a model describing the conditional distribution given in Eq. 3.4 makes it unnecessary to fit the residual noise as done with the TDNN in the previous section. Particles will still be used to simulate the accumulated noise, but instead of the adding a noise sampled from a fitted noise distribution, the next predicted state will be sampled directly from the Gaussian mixture model. This should result in a better residual noise representation, as the noise will depend on the sunspot activity, and not be constant. For example a Gaussian distribution used to represent the residual noise will not depend on the inputs to the network. It only depends on two constants. Including the inputs in the noise model, should give a better model for the noise.

Figure 3.6 gives the results for the sunspot forecasts using the Gaussian mixture model. It seems that the performance is a little worse for the GMM compared to the ANN.

For $d = 1$ and $d = 3$ the 90% percentile intervals stay mostly above zero, which is good, as this reflects the real data. This is not the case for $d = 11$, but this will be investigated later in this section.

In Figure 3.7 the placement of the GMM components is given. With $d = 3$ lags the GMM will have four dimensions, because the model output is also included in the joint probability function. The points in the scatter plots are all located in the positive quadrant I. There are several small clusters close to origin, as to model the abrubt change from high point density to none on the other side of the axis. In this way, the time-delay GMM model will learn that there is no negative sunspot activity, and this will be reflected in the particle boundaries.

As seen with the TDNN the noisy output stays within a reasonable area, but here it is the conditional distribution that maps the particles into a reasonable area.

For $d = 11$ the 90% percentile noise interval goes below zero, see Figure 3.6. The upper boundary is also high at the end of the forecast compared to what has been seen in the characteristics of the sunspot data. This has something to do with the nature of GMM. Each cluster in the GMM has a center and a covariance matrix. The covariance matrix has a size proportional to the square of the dimensions. The cluster center and covariance are calculated from the from the samples, which are estimated to belong to each cluster. If there are few samples compared to the dimension, then it can be problematic to determine the covariance matrices, and they might become singular. Effectively the covariance

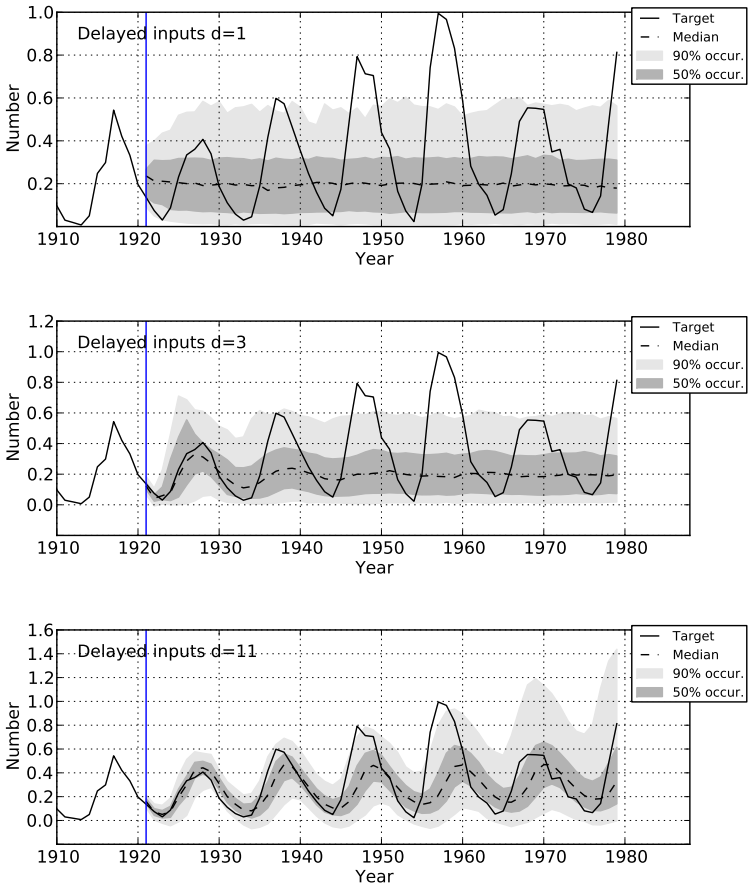

Figure 3.6: A time-delay network using a GMM for sunspot prediction. 1000 particles are used to simulate the accumulated residual noise from the probability density function obtained from the conditional distribution from the GMM model.

matrices are calculated only from a subset of the whole dataset, depending on the cluster mixing coefficient. A covariance has $D(D+1)/2$ adjustable free parameters, where D is the number of dimensions used. If we have $d = 11$ delayed inputs then the joint distribution will have $D = d + 1 = 12$ dimensions due to the output. This will give 78 free parameters for each covariance matrix. At this d value there are only 269 samples available to estimate all the covariance matrices. There are also additional free parameters in the model: the cluster means and mixing coefficients. This could become an issue depending on the

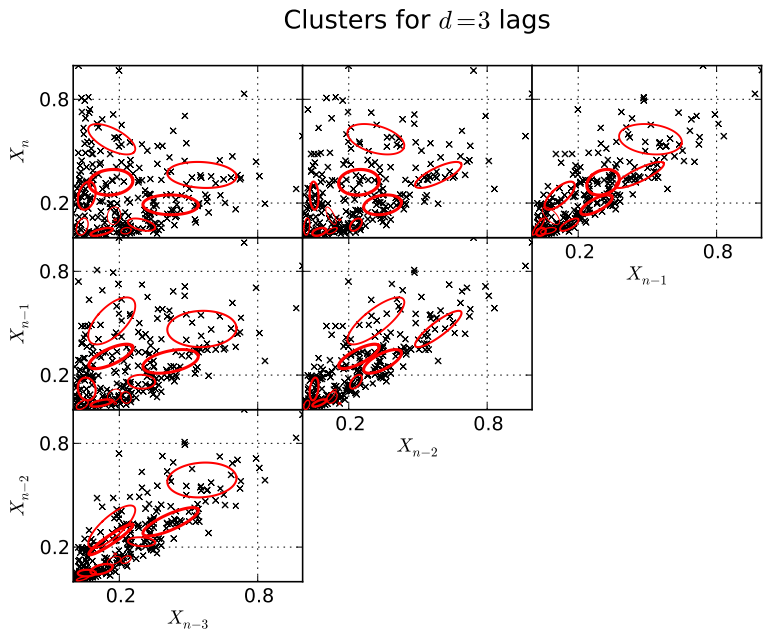

Figure 3.7: The GMM clusters component placement in plots of two dimensions for the case of $d = 3$ lags.

number of lags used in the ship propulsion model. However as mentioned earlier, a first order Markov approach will be taken, where $d = 1$ is used, and the data set is very large compared to the number of features, so this should not be a problem. If the number of lags where to be increased, dimensionality reduction methods, such as the Principal Component Analysis (PCA), could be used to tackle this problem. Another option is to constrain the covariance matrices, so that the number of free parameters is less, or use a Bayesian approach. [7]

Figure 3.8 show the case with $d = 11$ lags, but now with the inputs to the GMM have been reduced to 4 dimensions using PCA. It seems like the residual is modelled better now, as the 90% percentile interval does not go below zero.

The GMM could also be used to model the residuals of the ANN based system, instead of simply using the Gaussian, Student's t, or similar distributions. As the residuals of the ANN system are Gaussian like, the GMM should be able to model it easily. The ANN model output could also be presented to the GMM, so that the predicted residual density will depend on the inputs to the ANN

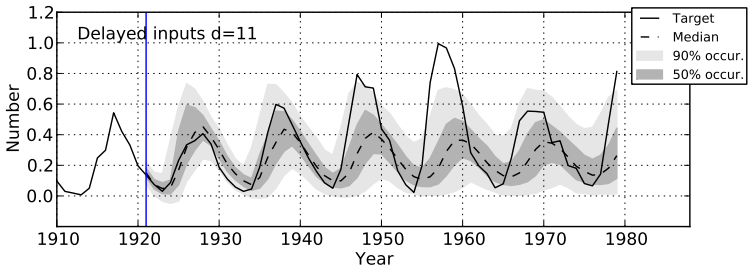

Figure 3.8: A time-delay network using a GMM for sunspot prediction. The difference from Figure 3.6 is that the dimensionality of the inputs to the GMM have been reduced to 4 using PCA, so that the covariance matrices can be more easily determined.

model as well.

	$x^{(5\%)}$	$x^{(25\%)}$	$x^{(50\%)}$	$x^{(75\%)}$	$x^{(95\%)}$
Train	0.02	0.07	0.20	0.33	0.58
TDNN $d = 1$	-0.06	0.14	0.28	0.43	0.63
TDNN $d = 3$	-0.06	0.14	0.28	0.42	0.60
GMM $d = 1$	0.01	0.06	0.18	0.31	0.56
GMM $d = 3$	0.02	0.07	0.20	0.32	0.56

Table 3.2: The percentiles for the training set and last results of the forecasts for the TDNN and GMM. Only $d = 1$ and $d = 3$ have been included for both models, as they seem to have evened out, and the percentile intervals are steady at the end of the forecast.

Table 3.2 gives some characteristics for the last year, 1980, in the forecasts for the TDNN and GMM. The table also gives the characteristics of the training set itself. The values are the percentile points, and the median of the particles, or samples in the case of the training set. The results are only given for $d = 1$ and $d = 3$, as these have evened out, and lost all information about the initial state. In this state the model should preferably predict the density of the sunspot training set values, as the models do not have any information about the state. As one might expect, the GMM numbers are quite similar to that of the training set. The TDNN number are somewhat different, but this is a expected due to the assumption that the residual have a stationary Gaussian distribution.

3.5 Alternative model approches

The models described until now have only dealt with observable variables or states. Models exist that can represent hidden states too. Hidden variables are also often referred to as *latent* variables. This class of *latent variable models* can be of interest in problems, where only a small proportion of the system is observable or the observations are distorted by noise. For example on a ferry with a roll-stabilizing system the state of this stabilizing system would be a latent variable if not measured. It would have an affect on the dynamics of the ship and its propulsion efficiency. In the sunspot problem, one could assume that there is some hidden process going on inside the sun, which produces the observable sunspots.

In the previous examples the state of the system is given by a set of observable variables, in the form of a number of previous sunspot values. The next state would then only depend on this observable set. In the sunspot problem there might be some hidden process that controls the generation of the sunspots. The process is hidden in the sense that there are no direct measurements of it. A system with hidden states is given by Eq. 3.5 and 3.6,

$$\mathbf{s}_{n+1} \sim p(\mathbf{s}_{n+1}|\mathbf{s}_n) \ , \tag{3.5}$$

$$\mathbf{x}_{n+1} \sim p(\mathbf{x}_{n+1}|\mathbf{s}_{n+1}) \ , \tag{3.6}$$

where s_n is a vector of hidden or latent variables. The next state of this system s_{n+1} only depends on the previous state s_n, similar to the first order Markov chain. A well known dynamic model with latent variables is the Kalman filter. The Kalman Filter can make a best estimate of the current state and its uncertainty, based upon the previous state and a measurement. This makes the Kalman Filter computationally cheap, as it only needs to look at the previous state and not a series.

The Kalman filter is a linear dynamic model, given by Eq. 3.7 and 3.8 [41, 19],

$$\mathbf{s}_{n+1} = \mathbf{F} \cdot \mathbf{s}_n + \mathbf{w}_n, \quad \mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{s}_0 \sim \mathcal{N}(\mathbf{S}_0, \mathbf{V}_0), \tag{3.7}$$

$$\mathbf{x}_n = \mathbf{H} \cdot \mathbf{s}_n + \mathbf{v}_n, \quad \mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{3.8}$$

where

- $\mathbf{s}_n$ is the hidden state vector
- $\mathbf{x}_n$ is the observations
- $\mathbf{s}_0$ is the initial state vector with mean $\mathbf{S}_0$ and with noise variance $\mathbf{V}_0$
- $\mathbf{F}$ is the state transition matrix
- $\mathbf{H}$ is the measurement matrix
- $\mathbf{w}_n$ is the process noise with variance $\mathbf{Q}$
- $\mathbf{v}_n$ is the measurement noise with variance $\mathbf{R}$

The main idea behind the Kalman filter is to model a hidden underlaying state that produces the sunspots. We only observe the sunspot activity, but infer the state variables. The process of inferring the state variables will not be discussed here, but an introduction to the concepts can be read in "Parameter Estimation for Linear Dynamic Systems" by Zoubin Ghahramani et al.[9]. The hidden states, observations, and the initial hidden state have an uncertainty or noise associated with them, which is assumed to be Gaussian. The noise added to the state vector gets accumulated over time. The variance of the Gaussian noise at some time index n on the measured signal can therefore be expressed as Eq. 3.9,

$$var(\mathbf{S}_n) = \mathbf{V}_0 + \sum_{i=1}^{n} \mathbf{Q} = \mathbf{V}_0 + n\mathbf{Q} \ . \tag{3.9}$$

The variance of the output noise grows linearly with time, and is not bounded. This is a drawback compared to the TDNN we have seen earlier and the Time-delay GMM model. The linear Kalman filter can therefore produce results, which are unrealistic for the system. For example as seen with the sunspot predictions, where the percentile intervals go below zero, stating a negative sunspot activity. Similar problems could arise with the ship system, for example with the speed of the vessel, fuel consumption, and so on.

A drawback is usability for a system with latent variables, is that we need to infer the latent variables before we can do forecasting. This is not so hard for a linear system like this, but could be problematic for nonlinear cases.

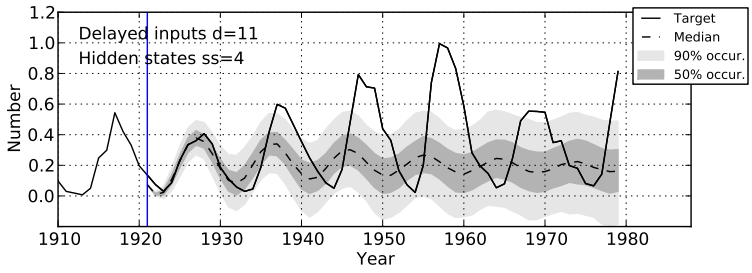

Figure 3.9: Predicting using a Kalman filter with 4 hidden state variables.

3.6 Comparison

Figure 3.10 gives a plot of the performance of the three model approaches versus the time index. The Kalman filter first two steps are quite poor, but after that it performs comparatively well in the beginning. It seems that the Kalman and GMM model occasionally have some very large errors at the same time. For example at the especially high sunspot activity in 1957 both model perform very poorly. The ANN however perform well here, but it could just be by chance.

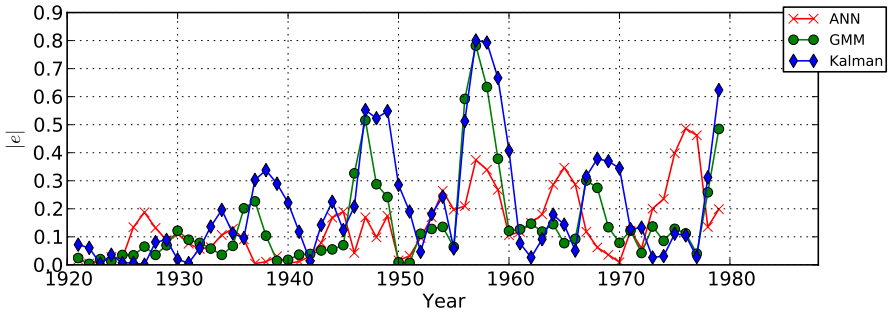

Figure 3.10: A comparison of the models using $d = 11$ lagged inputs. The error is the absolute difference between the correct value and the model output.

3.7 Conclusions and remarks

In this Chapter three types of networks capable of forecasting have been presented: TDNN, a GMM based network, and the Kalman filter. In the next chapter the TDNN and GMM network will be applied to the ship propulsion efficiency problem. The Kalman filter will not be studied further. It was included in this Chapter, so as to present it as a direction of further possibilites of further studies. The concept of including unobservable variables in the model is attractive. When installing a data collection system onboard a ship, there might be several reasons for some important signals not to be measured. For example due to costs, technical reasons, or oversight.

Chapter 4

Dynamical Modelling For Ships

In the previous chapter it was shown how time-delay networks could be used to forecast the sunspot activity. These methods will be extended and modified in this chapter, so that they can be used for forecasting the dynamical behavior of a ship.

The structure of this chapter is as follows. Section 4.1 extends the model that was used for the sunspot problem, so that it is suitable for the modelling ship time series. Section 4.2 examines the ship speed response to a rectangular propeller pitch signal. This example was presented at the COMPIT'11 conference, see Appendix B. Section 4.3 use the models on real data. The TDNN and a GMM models are set to follow the ship behavior stored in the data, and the results are compared to the real ship's response.

4.1 Ship Time Series Modelling

The ship time series are more complex than the sunspot time series. In the ship times series the sequences will consist of vectors with many different types of measurements. The entries in each sequence vector will be divided up into

control variables and observable state variables. How the entries are divided up depends on the application. However in general, it is assumed that the controls are set by the user, and are not affected by the state of the ship. The state variables in window n are denoted by $\mathbf{S}_n$ and the corresponding control variables by $\mathbf{U}_n$. One problem with the data set, is that we do not actually measure the controls used by the crew. For example we do not measure the requested propeller pitches, but only the actual propeller pitches, which will vary due propeller loading. The actual pitch will be treated as a control, even though this might be somewhat wrong. Future data sets for propulsion modelling should be aimed at having this information as well.

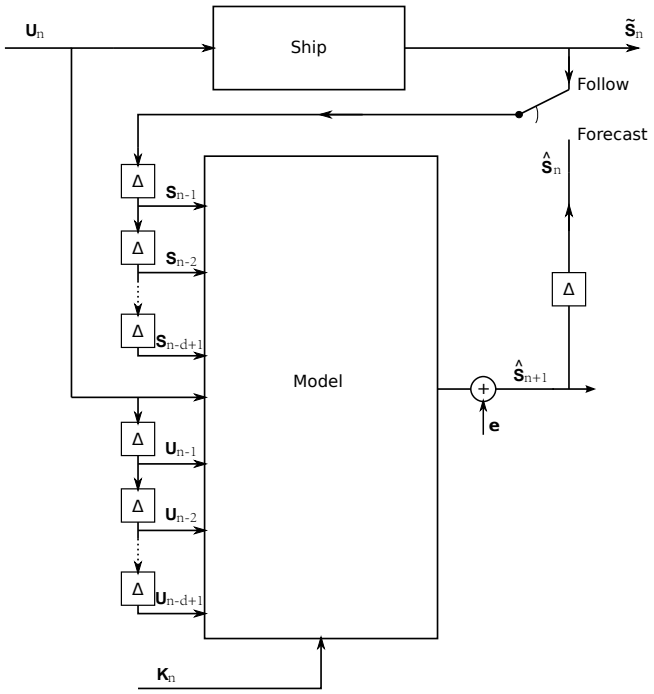

Figure 4.1: A time-delay network that has been extended to accommodate for modelling the ship propulsion problem. The system is based on the setup given in Figure 3.1, and basically works in the same way. The difference is that now the inputs to the model have been split up into controls, states, and constants. The model predicts the next state for the ship based on a sequence of historic values of these. The constants, $\mathbf{K}_n$, are measurements, which are assumed not to change for a long period of time.

Another way the sunspot time series and ship time series differ, is that the ship time series consists of several sets of sequences, instead of only one sequence,

assuming that a voyage or a leg in a trip is considered a sequence itself. Some constant variables will be generated for each of these sequences. The purpose of these are to hold some information about ship in a certain state. For example it is convenient to measure the loading of the ship in the harbor, where the ship has little speed and the waves are more limited than out on the sea. For the case of Smyril the constant inputs to the model are generated just after the departing berth. Here there typically is a small window where the ship is quite still, and the initial draft and trim at low speeds can be measured, which is related to the loading of the vessel. In the article in Appendix A the initial draft is studied. Here the number of cars on the deck and the bunkering of heavy fuel oil can be seen on the static draft measurements. The constant variables will be called $\mathbf{K}_n$. If no index is given, they are considered constant for the problem at hand. In fact the constant inputs $\mathbf{K}$ are just a special case of controls, which do not change at all, or very slowly compared to the other features of the system.

Another reason for introducing the constant inputs, is to give the model the ability to hold some ship state information over a longer duration. In Chapter 3 it was shown how the state variables where blurred over time due to the accumulated residual errors. It was concluded that the state variables are not a suitable way to hold states that are quite certain and do not change over time.

Figure 4.1 show a modified version of the time-delay network given in Chapter 3. This model has been extended with the ships states, controls, and constant inputs. The box denoted "ship" in Figure 4.1 is often called the plant in control applications. The illustration has been inspired by an illustration in "Fundamentals of Artificial Neural Networks" by Mohamad H. Hassoun [12].

It was mentioned in Section 3.3 that an order $d = 1$ Markov model would be used for modelling the ship's dynamics. At first this might seem strange, as the $d = 1$ models performed so badly on the sunspot data. The reason for why an order $d = 1$ Markov model might perform well on the ship data, is that the ship time series do not likely oscillate in the same way as the sunspot time series did over time. It they where to oscillate, it would be most likely due to the waves. With $d = 1$ the model will loose its ability to remember if the signals are increasing or decreasing. This is not so good, and work should be aimed at testing dynamic ship models with $d = 2$ or higher order in the future.

A difference between the sunspot and ship forecasting networks in Figures 3.1 and 4.1, is that there are controls and constant signals in the extended ship network that control the behavior of the output, the next predicted state. One might consider the control and constant signals as references for the model, so that it knows what a reasonable output is under the circumstances. So it is not likely that the output of the forecasting network will become wholly blurred as seen in the sunspot examples.

4.2 Pitch Experiments with a TDNN

The dynamic model presented in the article given in Appendix B will now be studied in more detail. It is based on a TDNN with the residual noise modelled with a multivariate Gaussian distribution.

The observable state vector, $\mathbf{S}_n$, consists of the following features:

- Speed through water (STW)
- Starboard and port level measurements
- Trim and draft
- True heading difference

And the control vector, $\mathbf{U}_n$, consists of the following features:

- Starboard and port propeller pitch
- Starboard and port rudder
- Longitudinal speed difference (STW-COG)
- Headwind and crosswind

The constant vector, $\mathbf{K}_n$, only consists of one feature:

- Initial starboard and port level

Compared to the size of the feature set used for regression this has few features. The reason for this is to keep it relative simple to start with. Some of the features used in regression might be useful here. For example the derivative of the pitches could be helpful, especially since the model is a $d = 1$ order Markov model. It would help the network see if the pitch is increasing or decreasing. On the other hand if $d = 2$ or higher were to be used, this feature might become less useful.

The initial starboard and port levels are the only constant features in $\mathbf{K}_n$. These are set at the start of the trip, and stored throughout the whole trip. This will help the system to remember how much the ship is loaded.

An intuitive example to demonstrate the models, is to change the pitch of the propellers, and let the models simulate the affects on the dynamic variables.

The dynamic state and the controls are set to a sample from the dataset, where the ship is quite still. The pitches are set to zero, and the model is allowed to settle in. This is why we see a little speed change in the beginning. Then the pitches are changed suddenly to value 80 pitch. There is no unit associated with the pitch, as it is not known what angle this measurement corresponds to. The sudden change is actually quite unrealistic, and one should actually study how the pitch changes occur in the data. However this approach gives an idea of the stability of the system.

It should also be noted, that one of the control inputs is the headwind, and that it has not been changed to reflect the speed change of the vessel. So the wind is following the vessel, so that the relative headwind always is zero.

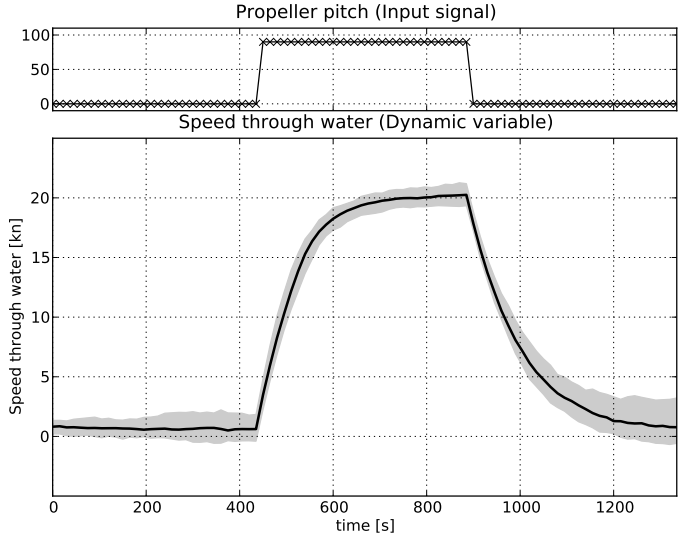

Figure 4.2: The response of the TDNN network to a rectangular pitch input. The upper plot show the propeller pitch signal, and the lower shows the forecasted speed through water response of the model. The marked area is the 90% percentile interval for the particles used.

The resulting plot is given in Figure 4.2. It can be seen how the ship speeds reacts to the pitch change. The boundaries of the forecasted 90% percentile interval for the speed goes below zero. The same was seen in the sunspot TDNN examples, but here is it not such a big problem, as there are cases where

the speed is negative. There are 1.6% of the 15 second windows sized training data, where the port pitch is positive and the ship has a negative speed through water. The number is 1.0% for the starboard pitch. Both propeller pitches are positive and the speed through water is negative for 0.9% of the data. It is interesting to note that the boundaries of the particles are narrower at cruise speed, where the data density is higher.

As a small experiment, one might initialize the model with the control, state, and constant vectors from the same sample from the data, and let the network forecast a few steps into the future. Depending on the situation, the state vector will likely change a little or much. This is due to the noise, window size, features used, and many other factors. It says something about the stability of the model in that situation, and is something that should be studied further. It might be the key to further improve the system.

4.3 Forecasting on real data

The correctness or performance of the dynamical models needs to be evaluated. For the regression model, this measure is quite simple to achieve, as one only needs to compare the estimated target value with the measured target value within a window. For the dynamic models this corresponds to comparing the predicted next state with the measured. The residual noise estimates are based on this difference using the validation data. But one might also want to get some proof, that the models are actually able to forecast the state of the system to some future point. One way of achieving this, is to let the network follow the controls of a dataset given some common start state. The initial dynamic state, controls, and constants are set to measured data, and then only the measured controls are fed to the system, while the dynamics are compared to the measured dynamics. This approach will be tried on the TDNN and GMM network.

The networks are a little different from the ones used in the previous section, Section 4.2. A state with the main fuel consumption is added, and two new constant signal, which give the initial draft and trim at the start of the journey.

The trips are identified in the dynamical dataset and shuffled, so that the trips occur in a random order. One third of the dynamical data set is removed for later use as test data. Of the remaining data 1/3 is used for a validation set. When training the TDNN the validation set is used for controlling the regularization of the model, and to estimate the parameters of the models used to represent the residual error of the model. For the TDNN a Gaussian multivariate noise profile is used, which is fitted to the residual of the validation part of the training data.

The GMM model selected with the best BIC score has 47 cluster components. The GMM maps the input directly into predicted states, and not the difference as the TDNN does.

The plot presented here are given using the test data sets, so that the time delay networks and the models for the residual error have not been presented with any information from the test data set. The training and test sets used for the TDNN and GMM network are identical.

To illustrate the difference between the model performances a trip will be studied at three places: the beginning, the middle, and the end. The model is initialized to measured data at beginning at each forecast. The TDNN results are shown in Figures 4.3, 4.4, and 4.5. The results for the same trips using the GMM network are given in Figures 4.6, 4.7, and 4.8. The plots show two state variables: the speed through water and the main fuel consumption. The measured correct values are given as dashed lines. The forecast is given as a solid line, and is surrounded by the 50% and the 90% percentile intervals.

The Gaussian noise seems to be somewhat underestimated in all three Figures 4.3, 4.4, and 4.5. The measured values often goes outside the 90% percentile interval of what the model expects the value to be. One reason for this is likely that the Gaussian noise is based upon a set of constant parameters, and does not take the inputs to the model into account. This problem was also discussed for the sunspot data, where it was seen as unrealistic sunspot numbers, such as negative sunspot numbers. Intuitively the models will perform poorer in complex situations, where the ship is in the harbor maneuvering, and best when the ship is cruising. The constant Gaussian noise parameters have to represent both situation. The residual noise at cruising speeds will dominate, as it represents a major portion of the data. The residual noise will therefore be underestimated for the complex situations. For the dynamical training set 88.4% of the windows have a speed through water, which is higher than 15 knots.

The GMM network seems to have the opposite problem when the ship is doing complex operations. The noise becomes so dominant, that the system looses all track of the fuel consumption in the beginning of the trip, see Figure 4.6. On the bright side, the 50% and 90% percentile interval also become very wide, and the measure fuel consumption is located within it. This reflects that the system is in a state of high uncertainty. One other possible factor for why GMM looses track of the fuel consumption is that it is not complex enough to express this part of the system. It is actually a piecewise linear model, which has been regularized used BIC. It would be interesting to see if a higher order Markov model would solve this, as it would be able track a rising or a falling fuel consumption.

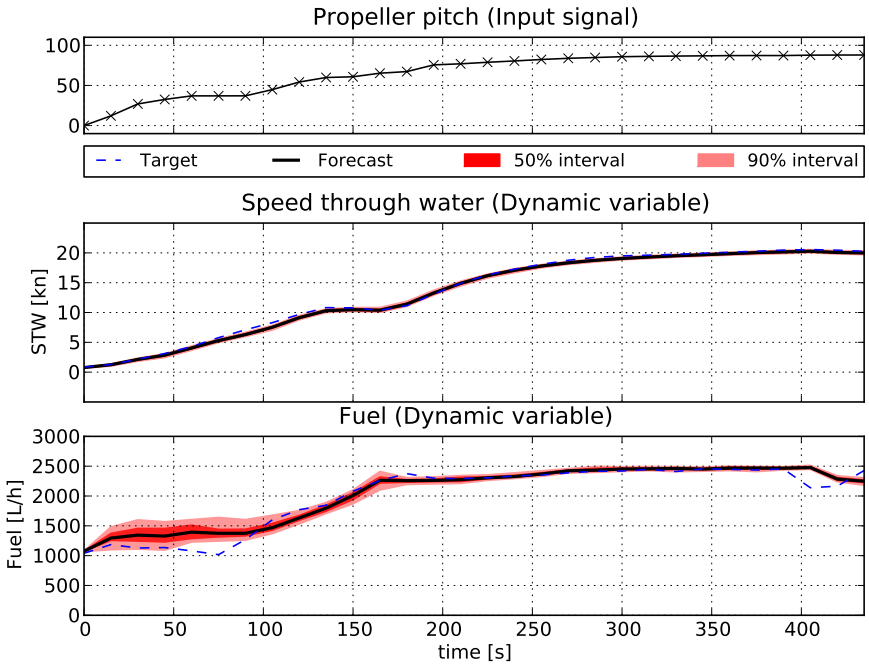

Figure 4.3: TDNN following real control values from the *beginning* of a selected trip. The change of the pitch is given in the top plot. The percentile intervals are very narrow for the speed. Maybe a little to optimistic, as the target seem to be outside the 90% percentile interval for some samples. The fuel intervals are broader, and reflects the difficulties in predicting the fuel.

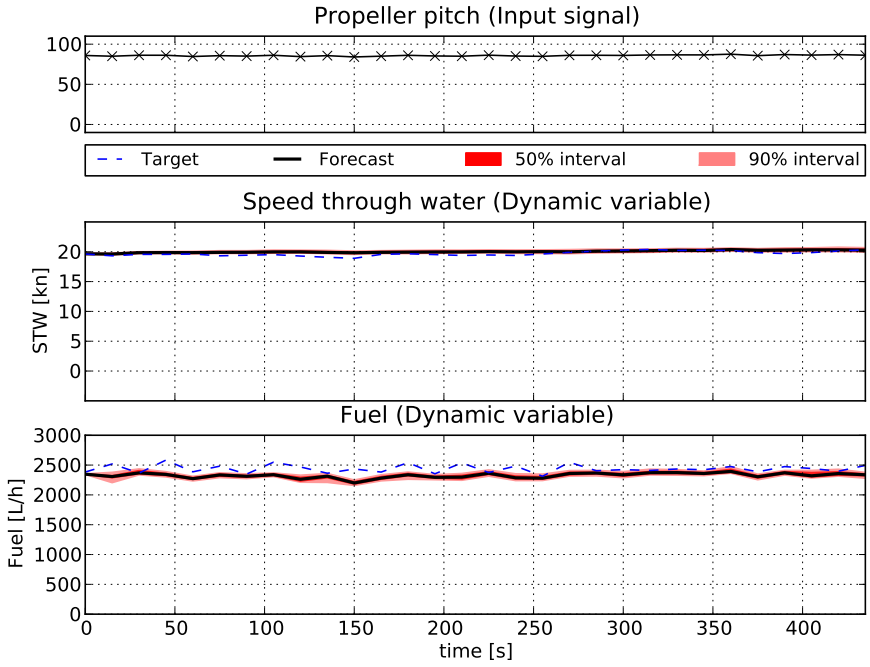

Figure 4.4: TDNN following real control values from the *middle* of a selected trip. Here the percentile intervals seem too optimistic: the speed and fuel targets are outside for both plots for several forecast steps.

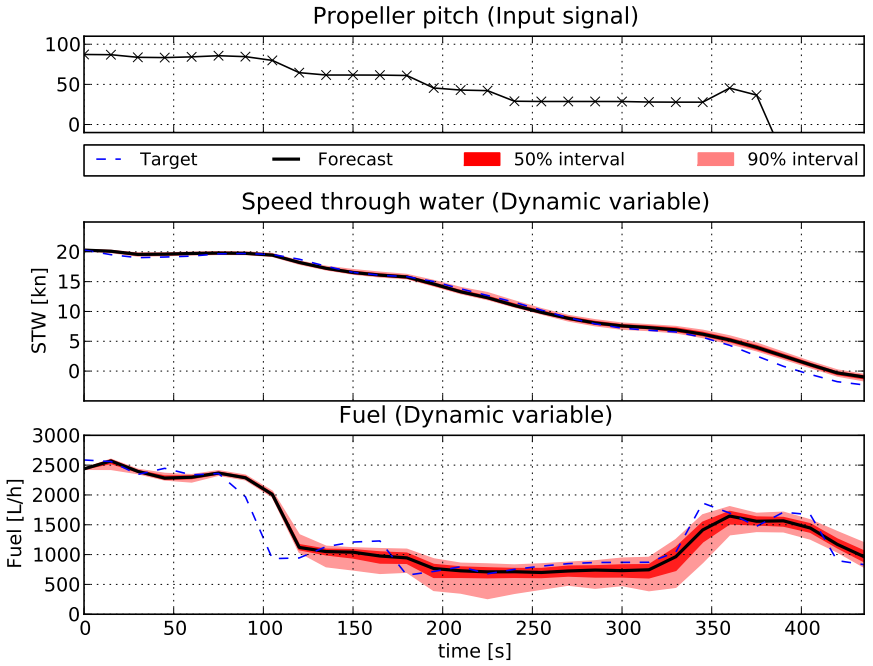

Figure 4.5: TDNN following real control values for the *end* of a selected trip. The pitch is reduced, as one might expect at the end of the trip. Some maneuvering in the harbor has not been included in the end of the trip. As seen with the two previous figures, the accumulated noise seems too optimistic.

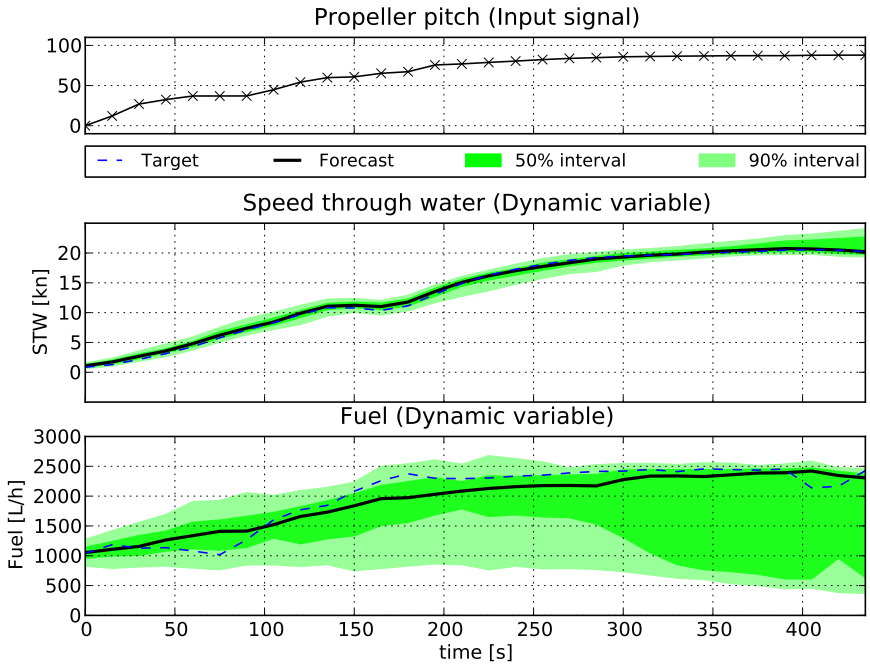

Figure 4.6: The GMM network following real control values from the *beginning* of a selected trip. Something interesting happens with the prediction for the main fuel consumption. Even though the predicted median is quite close to the measured target, the 50% and 90% percentile interval goes down to a very low fuel consumption. This reflects that many of the particles loose track of the fuel consumption. This does not happen for the speed through water to the same degree.

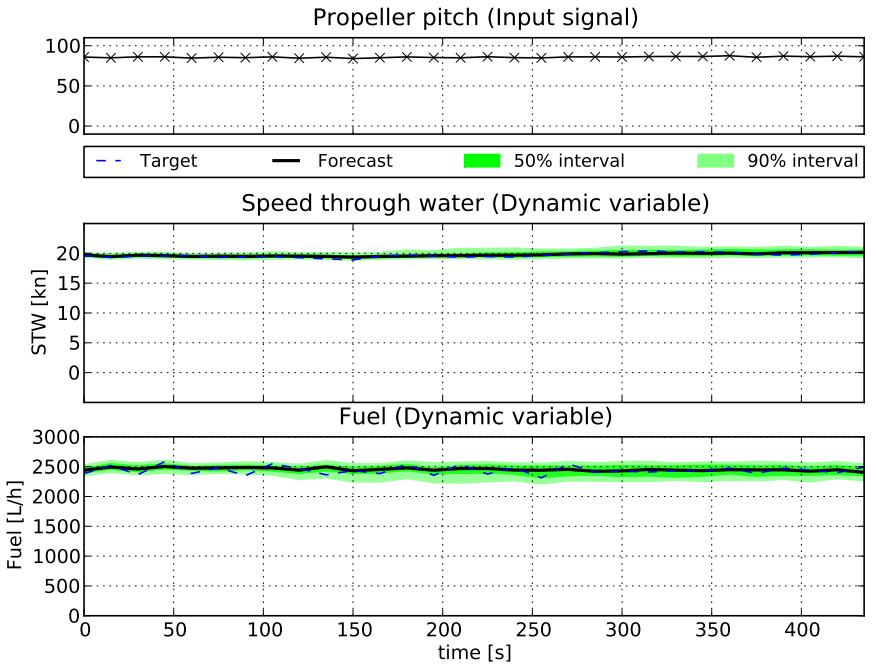

Figure 4.7: The GMM network following real control values from the *middle* of a selected trip. The percentile intervals for the GMM seem much broader than those seen with the TDNN with multivariate Gaussian noise.

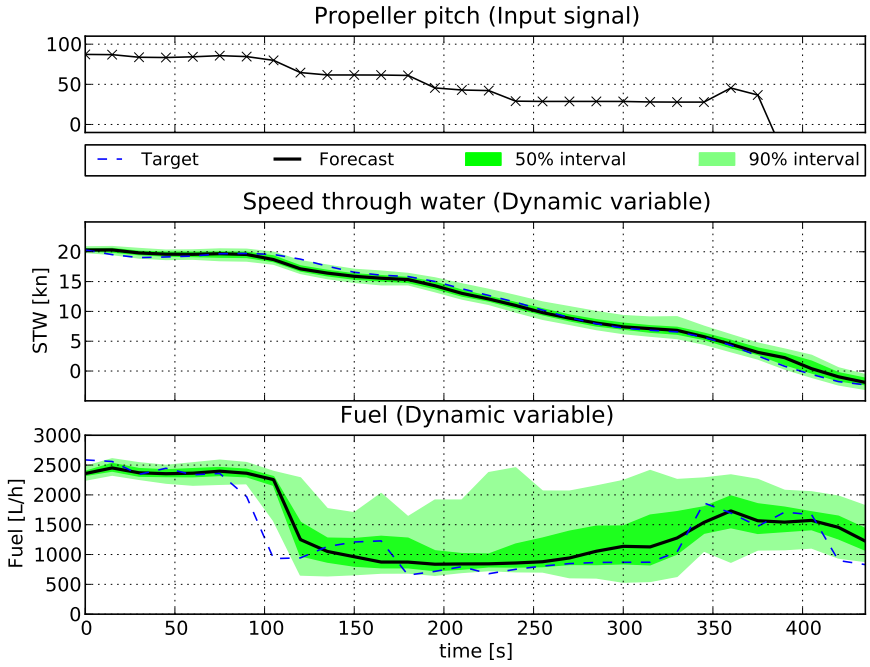

Figure 4.8: The GMM network following real control values from the *end* of a selected trip. The 50% and 90% percentile intervals become very broad here, but maybe not as bad as seen in for the beginning of the trip.

To get a final assessment of the percentile intervals of the two models a test has setup. The models are initialized 1000 times at random start points in the test data time series, and set to forecast 30 steps forward corresponding to 450 seconds. The number of times the forecast is outside the 50% and 90% percentile intervals are counted. The summary of the test is given in Table 4.1. The results are more consistent for the GMM network. The number of the measurements inside the percentile intervals correspond well with the interval range. The the GMM network might be a little pessimistic, because there is a little tendency for the results being a little higher than the values given by the percentile intervals. The results for the TDNN with the multivariate Gaussian noise seem not to work so well. These are similar to the results obtained with the sunspot examples. These results are very important, because they state that the GMM percentile intervals can be trusted under this setup.

	TDNN	GMM network
Speed inside 50% percentile interval	25.5%	60.3%
Fuel inside 50% percentile interval	25.0%	55.4%
Speed inside 90% percentile interval	55.7%	93.2%
Fuel inside 90% percentile interval	53.7%	92.4%

Table 4.1: A summary of how the percentile intervals perform given random initial start points in the test data. The numbers for the models should correspond to the percentile interval values.

4.4 Models with Hidden States

There are scenarios in ship propulsion modelling where a model with latent state variables could be beneficial. For example it is quite common that ferries have anti-rolling systems to improve the passenger comfort. Some of these stabilizing systems have wings mounted on the hull, that can be controlled generate hydrodynamic thrust to reduce the rolling. The generated thrust will affect the propulsion efficiency of the vessel, as it add resistance when active. If the state of the stabilizer is not measured, it will add an unobserved state to the system. A model it latent variables will be able to handle this, as it will discover based on measurements from the system, whether the stabilizers are on or off, and adjust the internal states accordingly. A system with no latent variables will not be able to account for this more than what is given in the observable states presented.

4.5 Conclusion

The forecasting models introduced for the sunspot time series have been extended for the ship time series. The models have be tested using real ship data, and are able to track the behavior of the ship. The GMM network approach is promising due to its good estimation of the accumulated residual noise. Models including latent variables seem like the natural next step in the evolution of forecasting models.

Chapter 5

Conclusion

This thesis has presented two approaches for ship propulsion efficiency modelling. The models have been based mainly upon statistical modelling approaches using collected operational ship data. One of the reasons for the successful implementations these models is the quality of the data. The quality of the data is one of the essential prerequisites for this approach to be a success.

All the regression models examined show good performance. However the Gaussian Process might be somewhat computationally demanding for an online system at this point in time. However there are many improvements that can be applied to this model, which have not been studied further here. So the model cannot be ruled out.

The dynamical models performed well. The GMM network forecasting approach performed well, and it was shown, that its percentile intervals could be trusted. This is a finding that is very useful for energy optimizing control and decision support applications.

The models can be improved much further, and there are several paths that can be followed from this point. One possibility is to apply the models to more complex scenarios. For example it would be relevant to try the system out on different ship types with different operational profiles. Further improvements mention in the text are:

- Improving the data collection.
- Study the feature extraction process in more detail.
- Study the window size with regard to applications, models, and features.
- Models with latent variables.
- Mix a physical model with the statistical model.

Appendix A

Statistical Modelling for Ship Propulsion Efficiency

Submitted to the *Journal of Marine Science and Technology*. Has been reviwed once, and is now accepted for publishing.

This paper presents a state-of-the-art systems approach to statistical modelling of fuel efficiency in ship propulsion, and also a novel and publicly available data set of high quality sensory data. Two statistical model approaches are investigated and compared: Artificial Neural Networks (ANN) and Gaussian processes (GP). The data presented is a publicly available full-scale data set, with a whole range of features sampled over a period of 2 months. We further discuss interpretations of the operational data in relation to the underlying physical system.

A.1 Introduction

A typical modern ship has several measurement devices that keep track of its operations, such as the vessel's speed, fuel consumption, weather conditions around the ship, and so forth. Using the right analytical tools, knowledge about the ship and its operations could be drawn from the collected data; thus making it valuable to the ship's owner, operators, and crew. This knowledge could be exploited both on-line on board the ship and off-line at the office. With an on-line system continuous control improvements could be made. This could be done in the form of control, advisory, or fault detection systems. Such systems would improve the efficiency and quality of the ship's operations.

The 1987 pioneering work "Marine Performance Surveillance with a Personal Computer" by Journée et al. [18, 17] describes an on-line data collecting system and a mathematical model to predict ship performance. This mathematical model is based upon physics and hydrodynamics formulas and the advisory system gives recommendations on trim, heading and speed for optimal performance regarding fuel consumption. The system also includes a voyage-planning module and estimation of fouling on the hull and propellers.

Today, more than two decades later, advances in computer, sensor and storage technology as well as in modelling have made significantly more accurate and thereby valuable systems possible [28, 21, 1]. However, although these advisory systems have been used onboard ships, it is safe to say that there still exist many open theoretical and practical questions before such systems will be routinely applied. We give an overview of relevant physical and/or statistical model based advisory systems in the following.

Ship resistance evaluation methods can be divided into four groups, from the traditional to the more advanced: Traditional and standard series methods, regression based methods, direct model tests, and computational fluid dynamics (CFD) [23]. The traditional, standard series, and regression based methods typically rely on a set of parameters that describe the hull, for example the wetted area, the Froude number, and so on. A well-know regression based approach to predict speed/power is the Holtrop-Mennen method [16, 15], which is widely applied in the initial design stage. Here the total resistance of the ship is subdivided into additive components, which are estimated based on data collected from model and full-scale tests. Other full-scale and model experiments have been carried out to investigate how different components affect the total

resistance of the ship, such as fouling and wind load [40, 4]. The Holtrop-Mennen methods have been applied for operational optimizations [21]. These initial design methods may sacrifice details for the sake of fewer parameters, and more robust results. Because of this sacrifice, these methods might not be very well suited for analysis of the ship's performance after it has been constructed. Computational Fluid Dynamics, CFD, can be used to estimate the hull resistance [36], but these calculations are computationally demanding, and typically take several hours to complete. So it is infeasible to use them onboard directly for optimization applications. It is however possible to do the CFD calculations beforehand for a given set of parameters. This technique has be used successfully for an trim advisory system [11].

In contrast to the physical models, statistical approaches only require knowledge about the ship and the physical set-up when it comes to the design of what sensory features to use in the system. The statistical model learns the relations between the measured signals from the collected training data. In the predictive machine learning approach the main agenda is generalization. This means the ability of the model to give sensible predictions for situations not identical to the what has already been observed in the training data. Generalization is not always the main concern [21]; interpretability may also be important for example in physical models. The main factors determining the predictive performance are the relevance of the sensory features, the amount of training data available and the choice and tuning of the complexity of the model to the task at hand. Hybrid physical and statistical models with a proper assessment of the predictive uncertainty of each component may in principle improve over a pure model by suitably interpolating between the two [21].

In a system that can optimize the fuel consumption, the two key quantities we need to be able to predict are the fuel consumption f and speed through water v. Ignoring route planning concerns the *fuel efficiency* is defined as the speed through water divided by the energy (fuel) consumed: $e = \frac{v}{f}$.

The ship's state is the set of variables that allow us to predict f and v. Depending on the application, a number of these quantities can be considered as *control variables* such as the trim, propeller pitch, RPM, i.e. these can be changed by the crew, while others like weather conditions or physical characteristics of the ship cannot. All relevant state variables are not measured - there can be several reasons for this, for example by choice, cost, or complexity. But arguably the most important factor in getting good model predictions is to have rich and accurate sensory data.

An instantaneous model predicts the current value based upon current measured signals. It therefore does not treat the control variables differently than the remaining variables. Using the model directly as part of an advisory system for

optimizing fuel efficiency on-line is problematic, because a change in a control variable may affect several variables, including inputs to the model and not only the outputs of the model. For example changing the trim could affect the way the autopilot controls the rudder, as the dynamics of the ship are also affected, which would result in an incorrect estimate of the speed and fuel consumption. So a temporal state-space model [10] for the dynamics of all variables is a more adequate model for this purpose. Nevertheless, the instantaneous model may be considered as an important first step towards a state-space or similar model since it actually models v and f which in the state space view are just two of the state-space variables. More importantly, the instantaneous model has important properties that can be useful for fuel efficiency as listed here:

Anomaly detection. The ship could for example have damages on the hull, which result in an reduction in the efficiency. This could be detected by comparing measured efficiency with the one predicted by the model and look for systematic bias. Here the model is used to detect an event that changes the efficiency of the vessel.

Benchmark modifications. It is difficult to compare the benefits of modifications to the ship. Here we have an event, where we would like to compare the efficiency before and after. The model is used as a benchmark before and after an event.

Fuel usage comparisons. Fuel usage comparisons for two periods can be tricky; there are several important parameters that affect the fuel consumption, which might differ between the periods. The weather conditions, tasks performed by the ship, and so on vary over time.

Long term drift. The prediction residual of the speed or efficiency may drift slowly over time, for example from fouling [40]

Data-mining state variables. Generally the model can be also be used to gain understanding of the ship's operation.

This paper describes a state-of-the-art systems approach to statistical modelling of fuel efficiency in ship propulsion. The two main contributing factors to the success of the system are the large data set of rich high quality sensory data and nonlinear machine learning regression models. The statistical model has been applied as part of a larger on-line system for optimizing the trim with respect to energy efficiency. The system advices the crew about optimal trim settings, based on current observations and historical operational data for the particular ship. We use this as an example application for the model; the model may, as discussed above, be used for other applications.

The rest of the paper is organized as follows: Section A.2 describes the data set and the various preprocessing steps, Section A.3 describes the machine learning methods used namely artificial neural networks and Gaussian processes, Section A.4 contains results and discussion and finally we conclude in Section A.5.

A.2 Data

The data is collected from a domestic ferry, MS *Smyril* in the Faroe Islands, owned by Strandfaraskip Landsins. The ship serves a daily route from the capital Tórshavn to the southernmost island Suduroy. The ferry sails two to three trips back and forth each day, where the duration of each trip is around 1 hour and 55 minutes. The ship is designed by Knud E. Hansen, and built on the IZAR shipyard, San Fernando, Spain (delivered in 2005).

A computer system and some additional hardware is installed onboard the ferry. This system collects the data from the ship, which the models will be based upon. The following measurements are stored by the system, which we will use in the models: Speed through water, fuel consumption, relative wind speed and angle, propeller pitch, rudder angle, trim angle, and the distance to the sea surface from two points on the ship. The draft and wave characteristics are derived from the sea surface distance measurements. The sea surface distance measurement devices are based on microwave technology. The devices are placed on the starboard and port side of the ship.

The data and code used in this paper is made publicly available, and can be found on `http://cogsys.imm.dtu.dk/propulsionmodelling/`. The raw and preprocessed data is available on the web page. The data spans a period of almost two months, February 16th to April 12th, 2010. We have made the data available in an attempt to encourage benchmarking within the ship propulsion field.

A.2.1 Data pre-processing

The samples are arranged into intervals or *windows*. The windows are non-overlapping and there is no gap between consecutive windows on a trip. If measurements are missing completely from a device in a window, the window is discarded. A device might have malfunctioned or stopped providing data for some other reason. Based on the samples within a window, a number of *features* are generated. The *feature extraction* process consists mostly of simple

mathematical operations such as taking the mean, variance, or derivative of an input. The mean, variance, and derivative features are calculated as

$$\Omega_{mean}(w) = \frac{1}{M} \sum_{n=0}^{M} x_n \tag{A.1}$$

$$\Omega_{var}(w) = \frac{1}{M} \sum_{n=0}^{M-1} \left(x_n - \Omega_{mean}(w)\right)^2 \tag{A.2}$$

$$\Omega_{der}(w) = \frac{1}{M-1} \sum_{n=0}^{M-1} \frac{x_{n+1} - x_n}{t_{n+1} - t_n} \tag{A.3}$$

where w is a window identifier, M the number of samples within the window, and x_n samples (n being its index) from the selected input signal within a window, and t the corresponding time stamps. The variance feature will express how much the signals vary within a window. The variance of the surface distance measurements will for example give the model an idea of the waves surrounding the ship. The derivative feature will tell the model if a signal is increasing or decreasing within a window. The choice of the window size depends on the application. We found that a window size of three minutes represent a good trade-off between robust estimation of data and time-scale for change in the variables for the trim optimization application. Using a window size of 3 minutes results in 9001 windows for the given data. Others have found a window size of 10 minutes to be suitable [28, 21].

A.2.2 Overview of sensor data and systematic effects

In this section we will provide some insight into the data and the underlying system. Two important features are the mean speed through water and mean fuel consumption for a window (the ones which we will predict). The minimum, maximum, mean, and standard deviation of the mean speed through water feature is 0.00, 21.59, 18.13, 4.09 knots, respectively; and 694.2, 2559.8, 2148.1, 355.8 L/h for the fuel consumption.

Figure A.1 shows a scatter plot of the mean fuel consumption against some selected features from the pre-processed data. A plus marker is used to indicate when the ship is accelerating, and a cross marker where the ship is decelerating. As expected there is a nonlinear relation between the speed through water

(STW) and the fuel consumption. The samples seem to lie on two paths in the speed through water and fuel plot, similarly in the pitch and fuel plot. If we look at the sample markers, we can see that this is due to acceleration and deceleration of the vessel.

In Figure A.2 we make a two dimensional visualization of the data and the influence of some of the variables using Principal Component Analysis (PCA). This can be used to map the data points down to two dimensions. Prior to PCA the data is standardized, so that the the transformed data has zero mean and unit variance. The Figure shows the projection of the data and unit vectors for some selected variables (port side pitch, trim, fuel consumption, and the speed) onto the two first principal axes, that is the two eigen vector of the data coaviarnce matrix with highest eigenvalues. The starboard pitch is not included, as it is very similar to the port pitch. The arrows of the speed, fuel, and pitch are also pointing in the same general direction. This is expected, as we know these to be very correlated. Different markers have been used for the samples to indicate their speed through water. We can see that the STW can not be clearly determined by using just these two principal components.

All of the signals supplied in the data set are common onboard most ships. However the microwave based level measurements devices are quite novel, though they have been used before [2]. With these devices it is possible to get information about the squat, trim, heeling, and draft of the ship, the waves generated by the motions of the ship, and sea waves around the ship. We will briefly look at them here, so as to give the reader an insight to the data. Much more work could be done interpreting these signals.

At the beginning of each trip it is possible to measure the *initial draft* of the vessel, before the vessel has gathered speed and squat and waves begin to affect the level measurements. This feature has not been used as an input to the models. We present it here to give an idea of the variation in the loading of the ship. The initial draft is plotted in Figure A.3*a* for the whole data set. The effects of the ship refuelling can be seen as two large jumps upwards in the draft measurements. The smaller faster variations are due to the amount of cars and cargo transported by the ship, as well as measurement noise.

Figure A.3*b* gives the data points sampled by the port side level measurement device versus the speed through water. The measurements are transformed so that they give the distance from the keel to the sea surface (draft). The port side radar is placed close to the bow of the ship, so the bow generated wave affects these measurements considerably. The bow wave height and length increases with the speed of the ship. This can be seen in the plot as an oscillating height measurements with increasing amplitude as the speed of the vessels increases. The measurements are also affected by trim and draft changes due to the speed of

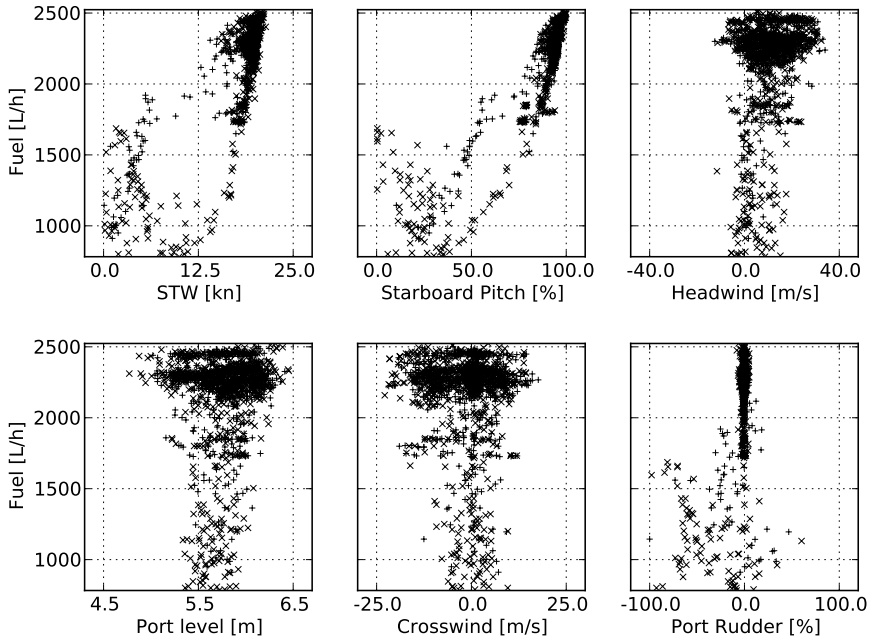

Figure A.1: Scatter plots of fuel consumption versus selected features for a part of the dataset. A plus marker is used to indicate when the ship is accelerating, and a cross marker where the ship is decelerating.

the vessel. If the functional form of these contributions to the level measurement is known, then their parameters could be fitted according to measured data. [6, 26] This has been roughly done with the line drawn through the data points in Figure A.3*b*.

A.3 Methods

We use supervised learning methods for building a regressional model for predicting the fuel consumption and speed from a set of measured features [3]. Formally the response (or output) variable, y, e.g. the fuel consumption, is modelled as

$$y = f(\mathbf{x}; \theta) + \epsilon \ , \tag{A.4}$$

where f denotes the model function depending upon parameters θ and explanatory (or input) variables $\mathbf{x}$. The residual ϵ is the part of the measured signal

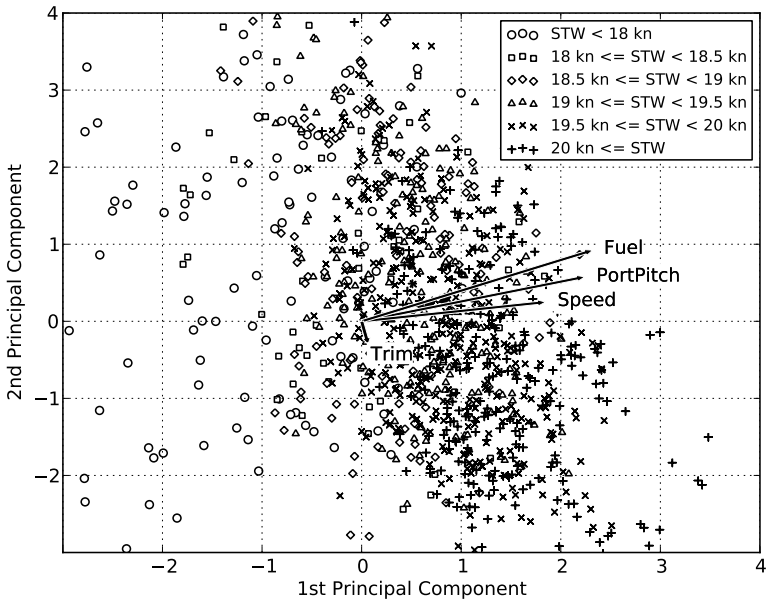

Figure A.2: Scatter plot in the two most important principal component axes. Samples have been marked according to their speed through water.

y not explained by the model due to noisy measurements and/or model shortcomings. The parameters θ should be learned (inferred) from a training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)|n = 1, \ldots, N\}$ of input-output pairs. We will use a probabilistic formulation assuming that the residual has a Normal (or Gaussian) zero-mean and σ_ϵ^2 variance distribution: $\mathcal{N}(\epsilon|0, \sigma_\epsilon^2)$. We will assume that measurements are independently identically distributed (iid) such that we can write the likelihood as

$$p(\mathbf{y}|\mathbf{X}, \theta, \sigma_\epsilon^2) = \prod_{n=1}^{N} \mathcal{N}(y_n|f(\mathbf{x}_n; \theta), \sigma_\epsilon^2) \ , \tag{A.5}$$

where $\mathbf{y}$ and $\mathbf{X}$ are shorthand for the vector of output and matrix of inputs, respectively. We will consider two state-of-the-art nonlinear models for f: artificial neural network (ANN) [3] and Gaussian processes (GP) [3, 35]. We briefly review these methods below including details for our use of the two methods, see [3, 35] for detailed general description. We take a predictive rather than descriptive modelling approach reporting the test (or generalization) performance on a *test set*. In all our experiments we use one third of the data as test set and the remaining two-thirds as training set. We investigate two different ways to make

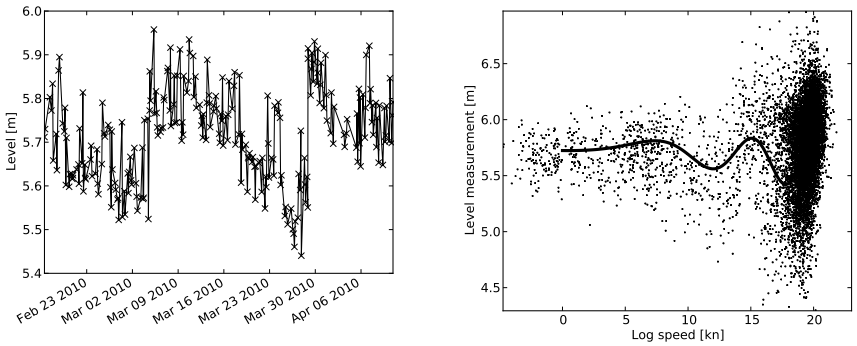

Figure A.3: **(a)** The initial draft measured at the beginning of each trip. **(b)** Measurements from the port side level radar, which is placed closed to the bow, plotted against the speed through water. The solid line gives an estimate of how the bow level is affected by the ship speed.

the training-test split. In the first approach test data is selected at random and in the second we minimize the training-test redundancy (or cross talk) coming from adjacent time-windows using complete legs as test and training data units, see Section A.4 for details.

A.3.1 Artificial Neural Network

We will use a feed forward neural network with two layers of adaptive parameters and bias units[1]:

$$f(\mathbf{x};\theta) = \tilde{g}\left(\sum_{j=0}^{M} w_j^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right) \quad . \tag{A.6}$$

The parameters of the model are the weights of the two layers $\theta = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)})$. Nonlinearity is achieved through the activation functions $\tilde{g}$ and g, which are usual taken to be sigmoid-functions such as hyperbolic tangent or the logistic function. In our case we use a linear output activation $\tilde{g}(a) = a$, and nonlinear activation for the hidden layer $g(a) = \tanh(a)$. The output of the network therefore becomes a linear combination of the nonlinear activation functions.

[1]The bias terms are included in a compact formulation by letting the summation start from zero and clamping both the zeroth input and the zeroth output from the hidden unit to minus one.

In order to control the complexity of the model we use regularization of the weights. This can be formulated in a number of ways. Here we view it as maximum a posteriori (MAP) estimation $\theta_{\text{MAP}} = \arg\max_\theta = p(\theta|\mathcal{D})$, where the posterior is given by

$$p(\theta|\mathcal{D}, \sigma_\epsilon^2, \sigma_\theta^2) = p(\mathbf{y}|\mathbf{X}, \theta, \sigma_\epsilon^2) p(\theta|\sigma_\theta^2) / p(\mathbf{y}|\mathbf{X}, \sigma_\epsilon^2, \sigma_\theta^2) \tag{A.7}$$

and $p(\theta|\sigma_\theta^2)$ is the prior distribution of the parameters. We take this distribution to be iid for weights and the prior for a single weight to be $\mathcal{N}(w|0, \sigma_\theta^2)$. We can formulate the MAP estimation problem as an equivalent cost function minimization problem by taking the logarithm and omitting terms not depending upon θ:

$$E(\theta; \lambda) = \frac{1}{2} \sum_{n=1}^{N} \left(f(\mathbf{x}_n; \theta) - y_n \right)^2 + \frac{\lambda}{2} \left\| \theta \right\|^2 \ , \tag{A.8}$$

where $\left\| \theta \right\|^2$ is shorthand for the sum of the squared elements of the weights and $\lambda = \sigma_\epsilon^2 / \sigma_\theta^2$. Optimization of this nonlinear cost function for constant hyper-parameter λ is performed by minimizing the error gradient found with back-propagation [3]. Model prediction is computed according to eq. (A.6) using the optimized weights which will be a function of both the training data, the hyper-parameters and possibly the initialization of parameters and learning rates. We discuss the effect of the latter in Section A.4. The learning of the hyper-parameter λ is done by 10-fold cross-validation. In k-fold cross-validation we split the training data in k approximately equal sized sets. We perform k training runs using in turn one of the sets for validation. The average test error over the k runs is proxy for the test error for a model trained on $N - N/k$ examples. We scan over a range of λ values and select the one with the lowest cross-validation error, see Section A.4.

A.3.2 Gaussian Processes

Placing a Gaussian process prior over functions $f(\mathbf{x})$ is an example of a non-parametric approach because no assumptions are made on the form (in terms of a distribution) of the input-output mapping. The artificial neural network model eq. (A.6), on the other hand, is a parametric model because it defines an explicit distribution for the mapping. The non-parametric approach allow us to define a very flexible class of distribution over functions which include but is not limited to known parametric choices such as the linear. Formally a GP is defined as follows: for any finite collection of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$, the joint distribution of functions defined in these points is multivariate Gaussian with

a mean value defined by the mean function $\mathbb{E}[f(\mathbf{x})] = m(\mathbf{x})$ and covariance defined by the covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$. We will take the mean function to be zero and choose the covariance function to be the squared exponential

$$k(\mathbf{x}, \mathbf{x}') = \sigma_{f0}^2 \exp\left(-\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{2\sigma_{fi}^2}\right) \tag{A.9}$$

where σ_{fi}^2, $i = 0, \ldots, d$ are the hyper-parameters of the model. The σ_{fi}^2, $i = 1, \ldots, d$ values are referred to as the squared length-scale for the input values. The length scale value essentially tells us how far we have to move in the input space in a particular dimension for the function values to become uncorrelated. This covariance function will produce smooth functions. A thorough discussion of covariance function and their properties can be found in [35]. We now turn to the discussion on how to make predictions and how to learn the hyper-parameters with the GP model. The GP formulation produces fully probabilistic predictions, i.e. for new input $\mathbf{x}^*$ we get a posterior distribution over output y^*: $p(y^*|\mathbf{x}^*, \mathcal{D})$. This means that we can not only give a point estimate using for example the mean value of the function but we can also quantify our uncertainty using the predictive standard deviation or some other similar summary. According to Bayes' theorem $p(y^*|\mathbf{x}^*, \mathcal{D}) = p(\mathbf{y}, y^*|\mathbf{X}, \mathbf{x}^*)/p(\mathbf{y}|\mathbf{X})$. This is nothing but the ratio between joint distributions of outputs for a set with $N+1$ and N examples. This probability can actually be calculated analytically because the model is fully Gaussian:

$$p(\mathbf{y}|\mathbf{X}, \sigma_\epsilon^2, \sigma_f^2) = \int p(\mathbf{y}|\mathbf{f}, \sigma_\epsilon^2)\, p(\mathbf{f}|\sigma_f^2)\, d\mathbf{f} = \mathcal{N}(\mathbf{y}|0, \sigma_\epsilon^2 \mathbf{I}_N + \mathbf{C}_N)\ , \tag{A.10}$$

where $\mathbf{f} = f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ is the vector of functions, $p(\mathbf{y}|\mathbf{f}, \sigma_\epsilon^2)$ is the likelihood eq. (A.5), $p(\mathbf{f}|\sigma_f^2) = \mathcal{N}(\mathbf{f}|0, \mathbf{C}_N)$ is the GP prior, $\mathbf{I}_N$ is the identity matrix of size N by N and $\mathbf{C}_N$ is the covariance matrix of training set with elements $C_{nn',N} = k(\mathbf{x}_n, \mathbf{x}_{n'})$. The predictive distribution is then simply

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \frac{\mathcal{N}(\mathbf{y}, y^*|0, \sigma_\epsilon^2 \mathbf{I}_{N+1} + \mathbf{C}_{N+1})}{\mathcal{N}(\mathbf{y}|0, \sigma_\epsilon^2 \mathbf{I}_N + \mathbf{C}_N)} = \mathcal{N}(y^*|\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*)) \tag{A.11}$$

with predictive mean and variance given by

$$\mu(\mathbf{x}^*) = \mathbf{c}^T (\sigma_\epsilon^2 \mathbf{I}_N + \mathbf{C}_N)^{-1} \mathbf{y} \tag{A.12}$$

$$\sigma^2(\mathbf{x}^*) = \sigma_\epsilon^2 + k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{c}^T (\sigma_\epsilon^2 \mathbf{I}_N + \mathbf{C}_N)^{-1} \mathbf{c} \tag{A.13}$$

and $\mathbf{c} = k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_N)$ is the covariance between the test point and the training points.

In principle we can learn the hyper-parameters by cross-validation as proposed for the artificial neural network model. However, eq. (A.10) gives us a simple

expression for the likelihood of the hyper-parameter. So we choose to learn the hyper-parameters by optimizing eq. (A.10) by conjugate gradients, see section A.4 for implementation details.

A.3.3 Performance measures

The accuracy of statistical prediction models may be quantified in terms of for example squared residuals of the model predictions on a test set and should be judged relative to baselines such as the error using the mean fuel consumption or the variance of the fuel consumption. The effect of single variables (for example the control variables) can also be investigated by comparing performance of models with and without these variables. A model which is successful in terms of significant improvement over the baseline can be considered a simulator of the ship in term of fuel consumption and speed. For regression models a convenient measure is the *root-mean-square error* (RMSE). It has the same unit as the model output variable, and has a similar form to the standard deviation. Given a data set containing N windows, the RMSE can be calculated according to equation A.14.

$$RMSE = \sqrt{\frac{\sum_n \|f(\mathbf{x}_n;\boldsymbol{\theta}) - y_n\|^2}{N}} \tag{A.14}$$

where f denotes the model function, and θ denotes a model parameter vector. The output of f for a given an input vector $\mathbf{x}_n$ is the predicted value by the model, and y_n is the value actually observed.

A.4 Results and Discussion

As mentioned in section A.3 the data set has been split into two sets, a training set and a test set. The training set is 2/3 of the data. This has been done by i) selecting random windows from the original set and ii) randomly selecting whole trips (sets of windows). The test sets are not used in any way before evaluating the performance of the models.

A.4.1 Model training

The weight decay parameter of the artificial neural network models are found using 10-fold cross validation as mentioned in section A.3.1. The training set is split into 10 approximately equal sized sets; this split is done window wise.

Figure A.4 shows the RMSE for the testing set using a different number of trips in the train set. Here the fuel consumption is predicted using the data split by whole trips. This is done in order to get an indication of how much data is required before the model can be used. Ten different trip sequences have been used - the variation resulting from this is illustrated by the boxplots. The shape of the learning curve is quite normal: a steep initial decrease followed by a slower (typically power-law) decrease.

The hyper-parameters of the GP are estimated using gradient ascent on the likelihood eq. A.10. The empirical results show sensitivity to initialization, indicating that a Bayesian estimation procedure might be more appropriate. Here we have opted for a robust initialization procedure, see below.

By studying the length scale hyper-parameters, σ_{fi}, in the GP kernel function eq.A.9, it is possible to get an insight into the relevance of each input. We have trained 10 GP models on the training data set, leaving 1/10 of the data set out for each model, this to see how sensitive the parameters are. The result is given in Figure A.5, where a box plot depicts the length-scale values found for each model input. The smaller the length-scale value, the more relevant the input is. A very large length scale value indicates that there is no correlation between that length scale value and the model output.

By examining Figure A.5 we notice that the length scales for both the starboard and port propeller pitches are short indicating that these are relevant when predicting the fuel consumption. In fact they are the two most important inputs. This does not come as a surprise, as these are the features from the data that most directly indicate the power used for moving the ship. The other inputs have a length scale starting almost an order of half magnitude longer than the propeller pitches. It is important to note that the relevance does not indicate the role of the variables in the underlying physical system.

A.4.2 Model Comparison

Table A.1 gives the performance results obtained for the models, along with the cross validation results. Even though it is not necessary to use cross validation

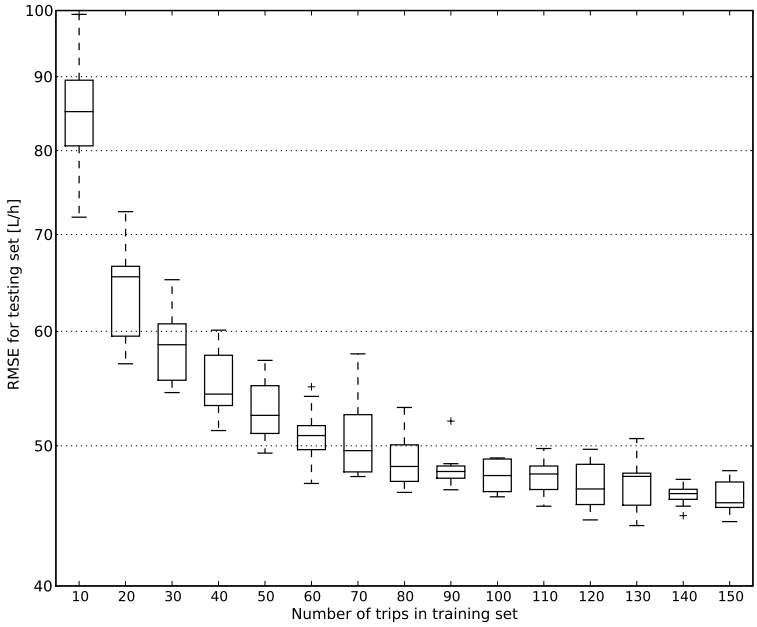

Figure A.4: Test set RMSE as a function of the number of trips in the training set. The fuel consumption is predicted using the data sets split up by whole trips. The error bars are obtained by ordering the way the trips are included in the data set used in 10 different ways.

with GP, we have obtained these results when making the box plots of the length scale hyper-parameters. The performance for the ANN is a little better than the GP in all the tests.

The cross validation results in Table A.1 show that all the ANN models have a low cross validation variance compared to their GP counterparts. This corresponds well with our experience training the models. The final GP models obtained were very sensitive to the starting conditions. This could indicate that the data allows for many model interpretations, which is seen as many local minima in the likelihood function. One should be aware of this problem when using the GP models, which can be circumvented to a degree by training several models with different starting conditions and by a good range within which to initialize the hyper-parameters.

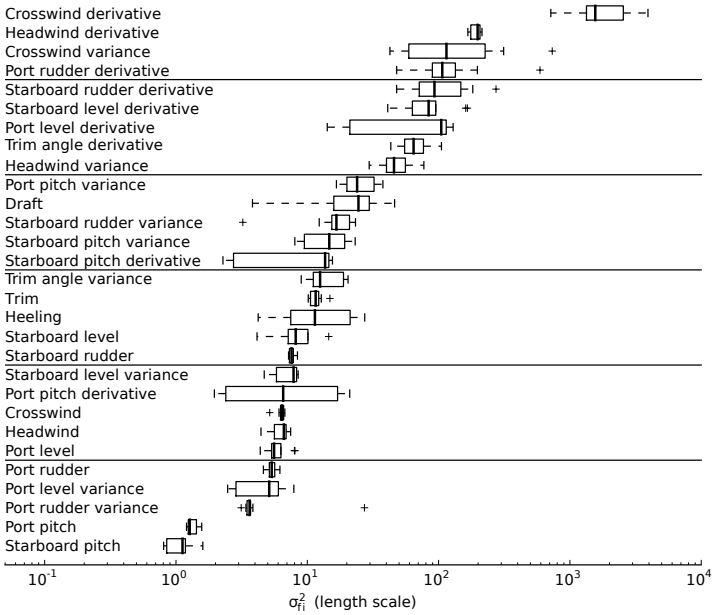

Figure A.5: Box plots of the length scale hyper-parameters from 10 GP models used for predicting the fuel consumption. The data split up by trips is used. The smaller the length-scale value is, the more relevant it is.

As an experiment, we have tried to remove all the features derived from the level measurement devices to see how it will affect the performance. These features are the draft, and the means, variances, and derivatives of the level measurements. This resulted in an root-mean-square error of 69.1 L/h, in the fuel estimate made by the ANN with the data set split by whole trips. This corresponds to a 47% increase in the error. Finally we have removed all the features except the starboard and port pitches, as these inputs are by far the most important for determining the speed and fuel consumption in this data. The obtained performance was an error of 104 L/h again using the ANN model with the data set split up by trips. This corresponds to a 121 % increase in the RMSE.

The performance is a little worse for all the models trained using the data set split up by trips as opposed to individual windows. This is expected, as one might assume that there is a temporal autocorrelation between the windows.

		Shuffeled windows		Shuffled trips	
		Speed [kn]	*Fuel* [L/h]	*Speed* [kn]	*Fuel* [L/h]
ANN	test	0.32	41.1	0.38	47.2
	x-val	0.32 ± 0.012	41.06 ± 0.76	0.39 ± 0.011	46.8 ± 0.75
GP	test	0.63	44.2	0.76	51.4
	x-val	0.63 ± 0.055	47.8 ± 2.55	0.75 ± 0.071	51.6 ± 2.87

Table A.1: Summary of model performance on the test sets (data not seen by the model before). There are two rows for each model. The test row gives the RMSE for the model trained on the whole training set, which is the final model obtained. And the x-val rows give the mean and standard deviation obtained from cross validation.

Windows within a trip are likely to be similar, so when the sets are split up window wise, similar windows from the same trip are likely to end up in both the training and testing sets. This will make it easier for the model predict the test values, as it has seen similar data before in the training process. The performance on the data split by windows is therefore too optimistic, and the performance results obtained by splitting the data sets up by trips is likely closer to the performance that will be obtained in a deployed model.

Splitting up the sets by trips gives us the ability to examine whole trips from the test set, because otherwise it would be likely that data points from a trip would end up in both the training and testing sets. Three selected trips are shown in Figure C.2, and the predictions obtained from the ANN and GP models. The first trip ends at around window number 40, and the second trip ends at around window number 75. It can be seen that the models are able to predict the changes within the trip, and that the target values lie within the confidence interval of the GP model. These plots give a qualitative impression of the models. In the first trip we can see that there is an constant offset error, which suddenly disappears at the end of the trip. This is most likely due to some part of the physical system from which we do not collect data. We hypothesize that the jump could be caused by the state of the stabilizers, from which we do not collect data. We can also see from the GP predictions, that the model is more uncertain at lower speeds. This is most likely because we have much less data collected at lower speed and because the ship is maneuvering in the harbor making the data more noisy.

A direct comparison with similar work [28, 21] is hard because the sensor data is different. Ref. [28] reports, as best result with an ANN, a mean relative error on *propulsion power* of 1.65%. Our results for ANN models are 1.50% for the *fuel consumption* using the data set split by windows and 1.67% if split by trips. Ref. [21] reports 0.65 knots RMSE for the speed and 60 L/h for fuel.

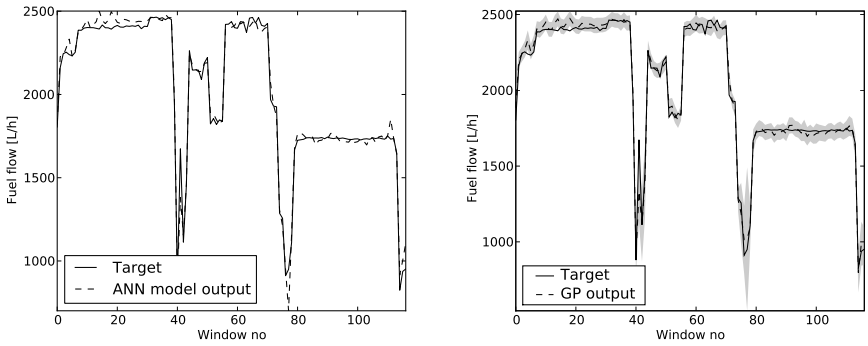

Figure A.6: These two plots give the model results for three selected trips from the test set (split trip-wise). The solid lines give the measure target values, while the dashed lines give the model predictions. The first plot **(a)** gives the output from the ANN model. The second plot **(b)** shows the output and 95% confidence interval obtained from the GP model. The confidence interval is given as the shaded area.

A.5 Conclusion

Two models for statistical modelling of fuel efficiency in ship propulsion have been presented. The models have their advantages and drawbacks, which we have discussed in Section A.4.2. The GP ability to quantify the uncertainty is a useful feature, especially for some applications, but its main drawback is that it scales poorly. The artificial neural network model has been used before in similar work[21, 28], while the Gaussian Process model has not been used in this context before. The performance obtained is similar to the previous work, however it is difficult to compare, because the data used is different. The difficulties comparing our results with previous work indicates that there is a need for some publicly available data, which can be used for benchmarking models and methods. The high quality sensory data set presented here can hopefully fill this gap, and be used as a common reference, or encourage others to make their data publicly available.

Our ongoing work within this area will focus on improving these models, especially with regard to the problem that several state variables are affected when changing one of them. This is a problem that for example can occur when using these models in a trim optimization application. For example advising a trim that is far from the current trim, will make the inputs invalid when the new trim is reached, and the model will give different results. It is plausible that a

state-space model or similar will be able to handle this problem.

Appendix B

A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions

---

Article presented at the 10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11) held in Berlin from 2-4 May 2011.

The conference preceedings have been published by the Technische Universität Hamburg-Harburg 2011, ISBN 978-3-89220-649-1.

A Machine-Learning Approach to Predict Main Energy Consumption under Realistic Operational Conditions

Jóan Petur Petersen, DTU Informatics, Lyngby/Denmark, joanpeturpetersen@gmail.com
Daniel J. Jakobsen, Decision3, Tórshavn/Faroe Islands, dj@decision3.com
Ole Winther, DTU Informatics, Lyngby/Denmark, owi@imm.dtu.dk

Abstract

We present a novel and publicly available data set of high quality sensory data collected from a ferry over a period of 2 months, and investigate state of the art machine-learning methods for prediction of main propulsion fuel efficiency. Neural networks are applied in both instantaneous and predictive settings. Performance results for the instantaneous models and examples and discussions of the practical advantage of the predictive models are given. The presented models have been successfully deployed in a trim optimization application onboard one of DS NORDEN's product tankers.

1. Introduction

Most modern ships have several measurement devices that keep track of its operations, such as the vessel's speed, fuel consumption, weather conditions, and so forth. Storing, analyzing, and acting upon this data could become a valuable asset for the ship owners, operators, and crew. In this article we present a freely available data set of data collected from a ferry, where the data has been used to develop the models used in a trim optimization application.

Already in 1987 the technical article "Marine Performance Surveillance with a Personal Computer" by *Journée (1987)* and *Journée et al. (2003)* described an on-line data collecting system and a mathematical model tuned by the collected data to predict ship performance. This mathematical model is based upon physics and hydrodynamics formulas. In the article by *Leifson et al. (2008)* model combinations of conventional physical models and artificial neural networks are tested. In the article by *Pedersen and Larsen (2009)* an Artificial Neural Network is used to predict the propulsion power. Both papers make use of collected operational data from the ship.

Ship resistance evaluation methods can be divided into four groups, from the traditional to the more advanced: Traditional and standard series methods, regression based methods, direct model tests, and computational fluid dynamics (CFD) *Molland (2008)*. The traditional, standard series, and regression based methods typically rely on a set of parameters that describe the hull, for example the wetted area, the Froude number, and so on. A well-know regression based approach to predict speed/power is the Holltrop-Mennen method, *Holtrop and Mennen (1982)*, *Holtrop (1984)*, which is widely applied in the initial design stage. Here the total resistance of the ship is subdivided into additive components, which are estimated based on data collected from model and full-scale tests. Other full-scale and model experiments have been carried out to investigate how different components affect the total resistance of the ship, such as fouling and wind load *Townsin (2003)*, *Blendermann (1996)*. The Holltrop-Mennen methods have been applied for operational optimizations *Leifson et al. (2008)*. These initial design methods may sacrifice details for the sake of fewer parameters, and more robust results. Because of this sacrifice, these methods might not be very well suited for analysis of the ship's performance after it has been constructed. Computational Fluid Dynamics, CFD, can be used to estimate hull resistance *Ruggiero et al. (2007)*, but these calculations are computationally demanding, and typically take several hours to complete. So it is infeasible to use them onboard directly for optimization applications. It is however possible to do the CFD calculations beforehand for a given set of parameters. This technique has been used successfully for an trim advisory system *Hansen and Freund (2009)*.

In contrast to the physical models, statistical approaches only require knowledge about the ship and the physical setup when it comes to the design of what sensory features to use in the system. The

statistical model learns the relations between the measured signals from the collected training data. In the predictive machine learning approach the main agenda is generalization. This means the ability of the model to give sensible predictions for situations not identical to what has already been observed in the training data. The main factors determining the predictive performance are the relevance of the sensory features, the amount of training data available and the choice and tuning of the complexity of the model to the task at hand. Hybrid physical and statistical models with a proper assessment of the predictive uncertainty of each component may in principle improve over a pure model by suitably interpolating between the two *Leifson et al. (2008)*.

In a system that can optimize the fuel consumption for propulsion, the two key quantities we need to be able to predict are the fuel consumption f and speed through water v. Ignoring route planning concerns the fuel efficiency is defined as the speed through water divided by the energy (fuel) consumed: $e = v / f$.

The ship's state is the set of variables that allow us to predict f and v. Depending on the application, a number of these quantities can be considered as control variables such as the trim, propeller pitch, RPM, i.e. these can be changed by the crew, while others like weather conditions or physical characteristics of the ship cannot. All relevant state variables are not necessarily measured - there can be several reasons for this, for example by choice, cost, or complexity. But arguably the most important factor in getting good model predictions is to have rich and accurate sensory data. An *instantaneous* model predicts the current value based upon current measured signals. It therefore does not treat the control variables differently than the remaining variables. Using the model directly as part of an advisory system for optimizing fuel efficiency on-line can be problematic, because a change in a control variable may affect several variables, including inputs to the model and not only the outputs of the model. For example changing the trim could affect the way the autopilot controls the rudder, as the dynamics of the ship are also affected, which would result in an incorrect estimate of the speed and fuel consumption. So a temporal state-space model, *Ghahramani and Hinton (1996)*, for the dynamics of all variables is a more adequate model for this purpose. We will not consider a complete state-space model with hidden states, but instead predictive model, which uses a non-linear neural network to capture the assumed Markovian dynamics of the system, *Weigend et al. (1990), Chakraborty et al. (1992), Svarer et al. (1993)*. This so-called *tapped-delay neural network* is not explicitly defined in terms of a model with hidden states like the traditional state-space model, but can never the less capture the essential part of the dynamics of the system. Ideally, the tapped-delay model represents the deterministic part of the dynamics. With careful regularization in order to avoid over-fitting, we may also use the residual error of the model on the training to fit an additive term in the model to represent the stochastic part of the model due to noise and/or un-observed information. The full model consisting of the deterministic and stochastic part allows us to make efficient non-linear propagation of predictions and uncertainty through time. The instantaneous model and the tapped-delay neural network, which we will focus on in this article, may be considered as an important first step towards a complete state-space model since it actually models v and f, which in the complete state space view are just two of the state variables.

2. Data collection

The data is collected from a domestic ferry, M/S Smyril in the Faroe Islands, owned by Strandfaraskip Landsins. The ship serves a daily route from the capital Tórshavn to the southernmost island Suduroy. The ferry sails two to three trips back and forth each day, where the duration of each trip is 1 h 55 min. The ship is designed by Knud E. Hansen, and built on the IZAR shipyard, San Fernando, Spain (delivered in 2005). Fig. 1 gives some data on the vessel.

A computer system and some additional hardware is installed onboard the ferry. This system collects the data from the ship, which the models will be based upon. The data is made publicly available, and can be found here: http://cogsys.imm.dtu.dk/propulsionmodelling/.

Length overall	135.00 m
Breadth moulded	22.70 m
Depth to main deck	8.10 m
Design draught	5.60 m
No. of private cars	970 m / 200 cars
No. of passengers	800 / 975 pax
Service speed	21 kn
Main engines	4 x 3360 kW (13440 kW)
Aux. engines	4 x 515 kW
Identification number	KEH 00072
IMO number	9275218

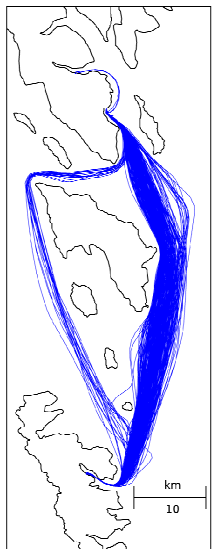

Fig.1: Information about the vessel Smyril, and the collected trips in the data set.

The data spans a period of almost two months, February 16th to April 12th, 2010. The map in Fig. 1 shows the routes taken by the vessel during the data collection period. We have made this data available in an attempt to encourage benchmarking within the ship propulsion field. Decision3 has provided the data for the project. To our knowledge, this is the first data set of its type that has been made publicly available. By releasing it to the community, we hope that it will be used for benchmarking similar models, and further work within this field.

The following signals are stored by the system: Port and starboard propeller pitch, port and starboard rudder angle, port and starboard level measurements, fuel density, fuel temperature, fuel volume rate, trim angle, longitude, latitude, speed through water, speed over ground, true heading, wind speed, and wind angle. Each signal is stored in a Comma Separated Values (CSV) file. For more details please refer to the homepage. All of the signals supplied in the data set are common onboard most ships. However the microwave based level measurements devices are quite novel, though they have been used before *Atwater (1990)*. With these devices it is possible to get information about the squat, trim, heeling, and draft of the ship, the waves generated by the motions of the ship, and sea waves around the ship. Much more work could be done interpreting these signals.

The system was setup to start storing data from the point when the ship starts to move and until it stops. Subsequently this might have been too restrictive, as the information just before and after a trip could also contain valuable information, which has not been stored. Smyril has a shaft generator, but it has not been used during the period of data collection. The engine has a constant RPM, and it has therefore not been measured.

If we look at the map in Fig. 1, we see that there are two short trips, where Smyril sails northward from Tórshavn. Here the ship is bunkering heavy fuel oil. The bunkering can be seen on the vessels draft. The variation in the routes taken is due to weather conditions.

3. Methods

3.1 Instantaneous/regression model

We use supervised learning methods for building a instantaneous/regression model to estimate the fuel consumption and speed from a set of measured features, *Bishop (2007)*. Formally the response (or output) variable, y, e.g. the fuel consumption, is modeled as

$$y = f(x;\theta) + \varepsilon, \tag{1}$$

where f denotes the model function depending upon parameters θ and explanatory (or input) variables x. The residual ε is the part of the measured signal y not explained by the model due to noisy measurements and/or model shortcomings. The parameters θ should be learned (inferred) from a training set $\mathbf{y} = \{(\mathbf{x}_n, \mathrm{y}_n) \mid \mathrm{n} = 1, \ldots, \mathrm{N}\}$ of input output pairs. We will use a probabilistic formulation assuming that the residual has a Normal (or Gaussian) zero-mean and σ_ε^2 variance distribution: $Norm(\varepsilon|0,\sigma_\varepsilon^2)$. We will assume that measurements are independently identically distributed (iid) such that we can write the likelihood as

$$p\left(\mathbf{y}|\mathbf{X},\theta,\sigma_\varepsilon^2\right) = \prod_{n=1}^{N} Norm\left(y_n|f(\mathbf{x}_n;\theta),\sigma_\varepsilon^2\right), \tag{2}$$

where $\mathbf{y}$ and $\mathbf{X}$ are shorthand for the vector of output and matrix of inputs, respectively. We will use an Artificial Neural Network (ANN), *Bishop (2007)*, a non-linear model for f. We briefly review this method below including details for our use of the method, *Bishop (2007)* for detailed general description. We take a predictive rather than descriptive modeling approach reporting the test (or generalization) performance on a test set. In all our experiments with the instantaneous model we use one third of the data as test set and the remaining two-thirds as training set. We investigate two different ways to make the training-test split. In the first approach test data is selected at random and in the second we minimize the training-test redundancy (or cross talk) coming from adjacent time-windows using complete legs as test and training data units, see Section 4.1 for details.

3.2 Tapped-delay neural network

While the instantaneous model only predicts the current target value, the tapped-delay neural network setup can predict future values.

A tapped-delay neural network model can be described as

$$\mathbf{Y}_n = f\left(\mathbf{X}_n, \mathbf{X}_{n-1}, ..., \mathbf{X}_{n-d+1}, \mathbf{w}\right) + e(n, \mathbf{w}), \tag{3}$$

where f is in this case is a artificial neural network that maps the seen sequence of sample vectors, $\mathbf{X}_n$, into a predicted future sample, w are the model's parameters often also called weights, and e is the error in the prediction. The error is only considered to be additive here. The number of previous samples vectors available to the network or steps is given by d.

We will make some small modifications to the tapped-delay neural network, as to adapt it more to this application. The output of the network is set to predict the next ship state, so that we want the output of the network to be $\mathbf{Y}_\mathrm{n} = \mathbf{X}_{\mathrm{n+1}}$. The controls are added as inputs to the model. The controls are considered to be known, so these are not predicted by the model. The controls are represented by a vector $\mathbf{U}_n$. The variable $\mathbf{X}_n$ will express the dynamic state of the ship. It will contain the ship's

speed through water, trim, draft, and so forth. The value from the current step, $\mathbf{X}_n$, is added to the right-hand side of Eq.(3). The argument for doing this, is that we suspect the next step, $\mathbf{X}_{n+1}$, to be quite similar to $\mathbf{X}_n$,, and therefore the model will only have to learn the difference between the steps. We will concentrate on the simplest tapped-delay neural network first, where d = 1. Applying these modification to Eq.(3) we obtain a new model given by

$$\mathbf{X}_{n+1} = f\left(\mathbf{X}_n, \mathbf{U}_n, \mathbf{w}\right) + \mathbf{X}_n + e(n, \mathbf{w}) . \tag{4}$$

3.3 Data pre-processing

The samples are arranged into intervals or windows. The windows are non-overlapping and there is no gap between consecutive windows on a trip. If measurements are missing completely from a device in a window, the window is discarded. A device might have malfunctioned or stopped providing data for some other reason. Based on the samples within a window, a number of features are generated. The feature extraction process consists mostly of simple mathematical operations such as taking the mean, variance, or derivative of an input. The mean, variance, and derivative features are calculated as

$$\Omega_{mean}(w) = \tfrac{1}{M}\sum_{n=0}^{M} x_n \ , \ \Omega_{\mathrm{var}}(w) = \tfrac{1}{M}\sum_{n=0}^{M}\left(x_n - \Omega_{mean}(w)\right)^2 \text{, and } \Omega_{der}(w) = \tfrac{1}{M-1}\sum_{n=0}^{M}\frac{x_{n-1} - x_n}{t_{n-1} - t_n} , \tag{5}$$

where w is a window identifier, M the number of samples within the window, and x_n samples (n being its index) from the selected input signal within a window, and t the corresponding time stamps. The variance feature will express how much the signals vary within a window. The variance of the surface distance measurements will for example give the model an idea of the waves surrounding the ship. The derivative feature will tell the model if a signal is increasing or decreasing within a window.

The choice of the window size depends on the application. We found that a window size of three minutes represent a good trade-off between robust estimation of data and time-scale for change in the variables for the instantaneous model. For the given data set using a window size of 3 min gives a total of 9001 windows. Others have found a window size of 10 min to be suitable *Pedersen et al. (2009)*, *Leifsson et al. (2008)*. The tapped-delay network example will have a much shorter length, as here we are interested in seeing the dynamic changes. The window size used for the tapped-delay neural network is 15 s.

Table I: Features extracted for the instantaneous model.

Description	Mean	Variance	Derivative
Speed through water *(target)*	√		
Fuel consumption *(target)*	√		
Trim	√	√	√
Port and starboard pitch	√	√	√
Port and starboard rudder	√	√	√
Heeling	√		
Draft	√		
Port and starboard level	√	√	√
Headwind and crosswind	√	√	√

Table I gives the features used for the instantaneous model tested. The features speed through water and fuel consumption are model outputs, while the rest are for inputs – there is a total of 29 inputs. The draft and heel are estimated from the trim and level measurements. For the tapped-delay neural network example, we will only use a small subset of these.

3.4 Artificial Neural Network

We will use a feed forward neural network with two layers of adaptive parameters and bias units[1]:

$$f(\mathbf{x};\theta) = \tilde{g}\left(\sum_{j=0}^{M} w_j^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i \right) \right). \tag{6}$$

The parameters of the model are the weights of the two layers $\theta = \left(\mathbf{W}^{(1)}, \mathbf{w}^{(2)}\right)$. Non-linearity is achieved through the activation functions $\tilde{g}$ and g, which are usual taken to be sigmoid-functions such as hyperbolic tangent or the logistic function. In our case we use a linear output activation $\tilde{g}(a) = a$, and non-linear activation for the hidden layer $g(a) = \tanh(a)$. The output of the network therefore becomes a linear combination of the nonlinear activation functions.

In order to control the complexity of the model we use regularization of the weights. This can be formulated in a number of ways. Here we view it as maximum a posteriori (MAP) estimation $\theta_{MAP} = \arg\max_\theta = p(\theta|D)$, where the posterior is given by

$$p\left(\theta \middle| D, \sigma_\varepsilon^2, \sigma_\theta^2\right) = \frac{p\left(\mathbf{y} \middle| \mathbf{X}, \theta, \sigma_\varepsilon^2\right) p\left(\theta \middle| \sigma_\theta^2\right)}{p\left(\mathbf{y} \middle| \mathbf{X}, \sigma_\varepsilon^2, \sigma_\theta^2\right)}. \tag{7}$$

and $p\left(\theta \middle| \sigma^2\right)$ is the prior distribution of the parameters. We take this distribution to be independent and identically distributed (i.i.d.) for weights and the prior for a single weight to be $Norm(w|0,\sigma_\theta^2)$. We can formulate the MAP estimation problem as an equivalent cost function minimization problem by taking the logarithm and omitting terms not depending upon θ:

$$E(\theta;\lambda) = \tfrac{1}{2} \sum_{n=1}^{N} \left(f(\mathbf{x}_n;\theta) - y_n \right)^2 + \lambda \|\theta\|^2, \tag{8}$$

where $\|\theta\|^2$ is the two-norm or sum of the squared elements of the weights and $\lambda = \sigma_\varepsilon^2 / \sigma_\theta^2$. Optimization of this non-linear cost function for constant hyper-parameter λ is performed by minimizing the error gradient found with back-propagation *Bishop (2007)*. Model prediction is computed according to Eq. (6) using the optimized weights which will be a function of both the training data, the hyper-parameters and possibly the initialization of parameters and learning rates. The learning of the hyper-parameter λ is done by *k*-fold cross-validation, where *k*=10. In *k*-fold cross-validation we split the training data into k approximately equal sized sets. We perform *k* training runs using in turn one of the sets for validation. The average test error over the k runs is proxy for the test error for a model trained on $N - N/k$ examples. We scan over a range of λ values and select the one with the lowest cross-validation error, see Section 4.

[1] The bias terms are included in a compact formulation by letting the summation start from zero and clamping both the zeroth input and the zeroth output from the hidden unit to minus one.

3.5 Performance measures

The accuracy of statistical prediction models may be quantified in terms of for example squared residuals of the model predictions on a test set and should be judged relative to baselines such as the error using the mean fuel consumption or the variance of the fuel consumption. The effect of single variables (for example the control variables) can also be investigated by comparing performance of models with and without these variables. A model which is successful in terms of significant improvement over the baseline can be considered a simulator of the ship in term of fuel consumption and speed. For regression models a convenient measure is the root-mean-square error (RMSE). It has the same unit as the model output variable, and has a similar form to the standard deviation. Given a data set containing *N* windows, the RMSE can be calculated as follows

$$RMSE = \sqrt{\frac{\sum_n \left\| f(\mathbf{x}_n|\theta) - y_n \right\|^2}{N}}, \tag{9}$$

where f denotes the model function, and θ denotes a model parameter vector. The output of f for a given an input vector $\mathbf{x}_n$ is the predicted value by the model, and y_n is the value actually observed.

4. Results

4.1 Instantaneous model results

As mentioned in section 3.4 the data set has been split into two sets, a training set and a test set. The training set is 2/3 of the data. This has been done by i) selecting random windows from the original set and ii) randomly selecting whole trips (sets of windows). The test sets are not used in any way before evaluating the performance of the models. The weight decay parameter of the artificial neural network models are found using 10-fold cross validation as mentioned in section 3.4. The training set is split into 10 approximately equal sized sets; this split is done window wise. Fig. 2 shows the RMSE for the testing set using a different number of trips in the train set. Here the fuel consumption is predicted using the data split by whole trips. This is done in order to get an indication of how much data is required before the model can be used. Ten different trip sequences have been used - the variation resulting from this is illustrated by the box plots. The shape of the learning curve is quite normal: a steep initial decrease followed by a slower (typically power-law) decrease.

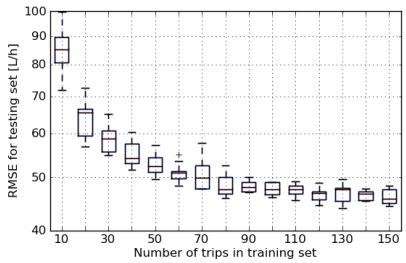

Fig.2: Test set RMSE as a function of the number of trips in the training set. The fuel consumption is predicted using the data sets split up by whole trips. The error bars are obtained by ordering the way the trips are included in the data set in 10 different ways.

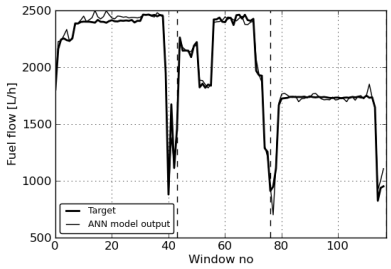

Fig.3: The output from the final ANN model and the corresponding target (true) values. The data points are from three selected trips from the test set (split trip-wise). The trips are separated by vertical dashed lines in the plot.

By splitting up the sets by trips, gives us the ability to examine whole trips from the test set, because otherwise it would be likely that data points from a trip would end up in both the training and testing sets. Three selected trips are shown in Fig. 3, and the predictions obtained from the ANN. It can be seen that the model is able to predict the changes within the trip. These plots give a qualitative impression of the models.

Table II gives the performance results obtained for the instantaneous models, along with the cross validation results, which are expressed as a mean value and a standard deviation. As one might have expected, the models perform a little better using the dataset where the windows are shuffled, due to the cross talk mentioned in section 3.1.

A direct comparison with similar work *Pedersen et al. (2009)*, *Leifsson et al. (2008)* is hard because the sensor data is different. *Pedersen et al. (2009)* report, as best result with an ANN model, a mean relative error on propulsion power of 1.65%. Our results for ANN models are 1.50% for the fuel consumption using the data set split by windows and 1.67% if split by trips. *Leifsson et al. (2008)* report 0.65 knots RMSE for the speed and 60 L/h for fuel. Using the mean relative error is a disadvantage to our model and data, because we have relatively many samples where the speed is low compared to the two other articles.

Table II: Performance results for the instantaneous ANN model

		Shuffled windows		Shuffled trip	
		Speed [kn]	Fuel [L/h]	Speed [kn]	Fuel [L/h]
Performance RMSE	test	0.32	41.1	0.38	47.2
	x-val	0.32 ± 0.012	41.06 ± 0.76	0.39 ± 0.011	46.8 ± 0.75

Figs. 4 and 5 show the histograms for the residuals for two modeled feature values. The distributions of the residuals have been approximated by a normal distribution and Student's t distribution. The Student's t distribution has longer 'tails' than a Gaussian. This makes the distribution much less sensitive than the Gaussian to the presence of a few data points which are outliers, *Bishop (2007)*. The probability density distribution for the speed and fuel residuals is narrower than the fitted normal distribution. Outliers often arise in practical applications either because the process that generates the data corresponds to a distribution having a heavy tail or simply through erroneous data.

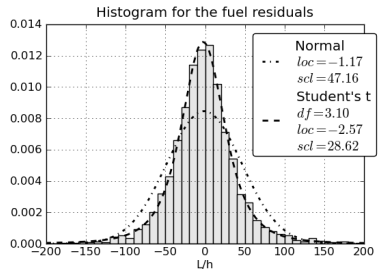

Fig.4: Histogram of the residuals for fuel consumption using the instantaneous model

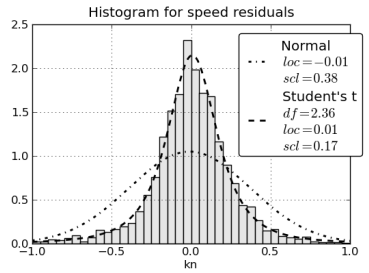

Fig.5: Histogram of the residuals for speed through water using the instantaneous model

4.2 Tapped-delay neural network model and residual modeling

We will keep the tapped-delay neural network example simple, and only include a few parameters in the model. The following control features have been used: Port and starboard pitch, port and starboard rudder, initial port and starboard level, difference between ground and speed through water, headwind and crosswind. The following dynamic variables have been used: Speed through water, port and starboard level, trim, draft, difference in heading. The model is trained on 2/3 of the data using a

range of weight decays values, and the model with the best performance based on the rest of the data is selected (validation set).

The predictive model is evaluated using Eq.(4). The noise, $e(n, \mathbf{w})$, will be simulated using a Gaussian and Student's t distribution fitted to the residuals on the validation data. We will use the univariate distribution here for each variable. The model can be initialized with some reasonable settings for the dynamic values, and the model's response to a given input can be found. Including the noise in the dynamic states will give an idea of the sensitive the system is. By examining the plots in Figs. 6 and 7 where the histograms for two of the dynamic variables are given, it is evident that the student's t distribution is a good representative for the residual noise.

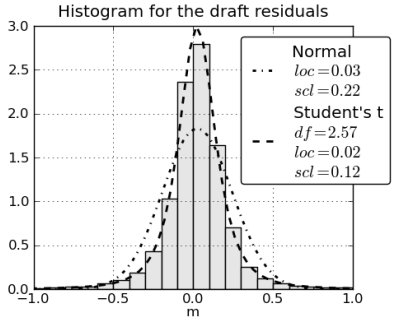

Fig.6: Histogram of the draft residuals using the predictive model

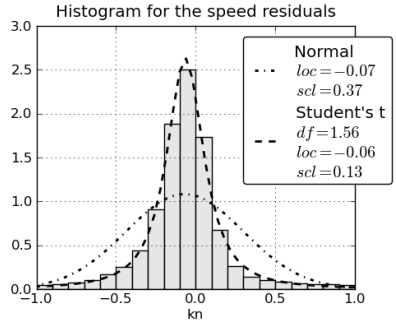

Fig.7: Histogram of the speed through water residuals using the predictive model

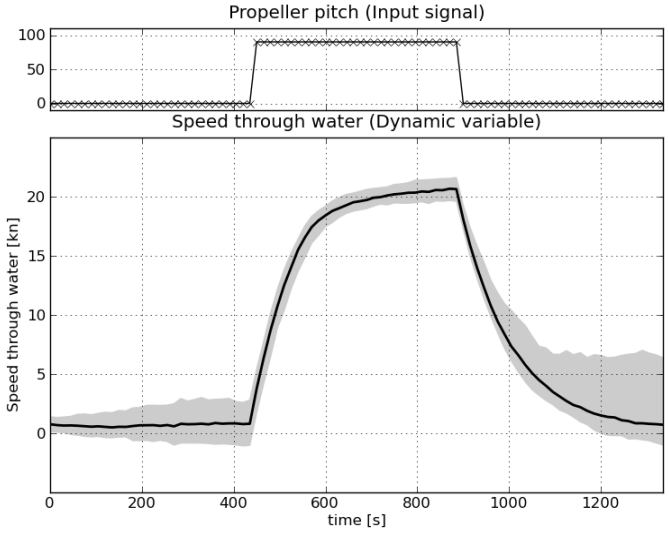

Fig.8: Tapped-delay neural network response to a (rectangular) step function for the propeller pitch. Both propeller pitches are set to the same value. The solid gray area gives the models sensitivity to added Gaussian multivariate noise.

As an example, we have generated 100 dynamic variable samples, and propagated these through the tapped-delay neural network for 90 steps, where each step is 15 seconds. The propeller pitches are changed from being 0 for 30 steps, to being 90 for 30 steps, and then 0 again for 30 steps. The control signal is drawn in the upper plot in Figs. 8 and 9, and the tapped-delay neural network's response is given in the lower plot. The samples have been ordered after the speed through water value, and the area between the 5 lowest and highest values is given by the solid gray interval in the plots. Fig. 8 gives the response from the tapped-delay network with added Gaussian noise, while the plot in Figur 9 gives the response with added noise from the Student's t distribution. If we look at the plots, we can see that the uncertainty in the speed prediction grows with time as the noise accumulates. It also seems like the noise is worse with lower speeds than higher speeds – a possible reason for this could be that we have much more data at the ship's cruising speed, and a smaller part of the data at lower speeds. The system also stops storing data just before the ship stops, so the system newer actually experiences the ship being totally still.

If used correctly this model approach gives us the ability to determine how a change in a control will affect all of the dynamics of the ship; for example how changes in propeller pitch would affect the trim of the vessel. This is a clear advantage compared to the instantaneous model.

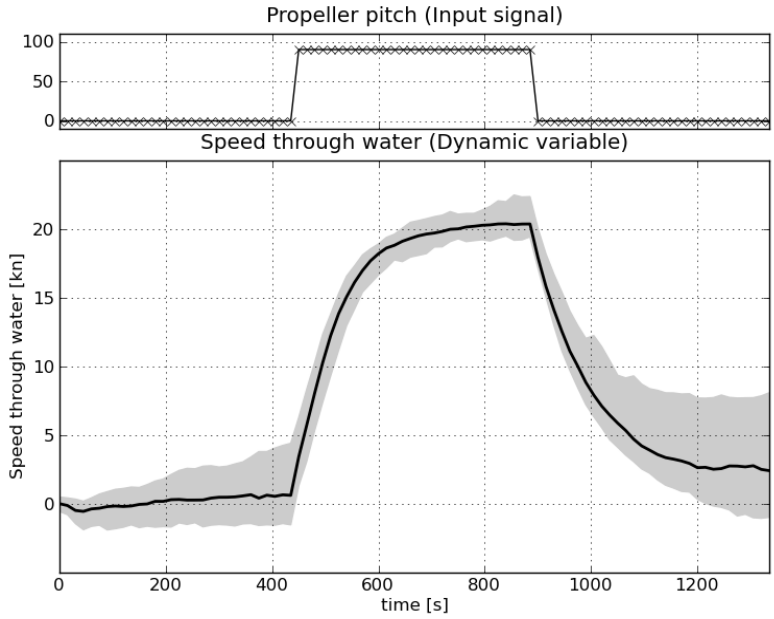

Fig.9: Using the same model and control signal sequence as in Fig. 8, but now the noise is drawn from the student's t distribution

5. Conclusion

A non-linear ANN model has been used for modeling of fuel efficiency in ship propulsion. The instantaneous model, using a artificial neural network model, has been used before in similar work *Pedersen et al. (2009)* and *Leifsson et al. (2008)*. The results obtained are similar to those of previous works; it is however difficult to compare these results, because the data used is different. Difficulties comparing our results with previous work might indicate that there is a need for some publicly available data, which can be used for benchmarking these models and methods. The high quality

sensory data set presented here can hopefully fill this gap, and be used as a common reference, or encourage others to make their data publicly available.

Our ongoing work within this area will focus on improving these models, especially with regard to the problem that several state variables are affected when changing one of them. This is a problem that for example can occur when using these models in a trim optimization application. For example advising a trim that is far from the current trim, will make the inputs invalid when the new trim is reached, and the model will give different results. It is plausible that a state-space model or similar may be able to handle this problem.

References

ATWATER, R. (1990), *Shipboard wave height sensor*, Tech. rep., Ship Structure Committee

BLENDERMANN, W. (1996), *Wind loading of ships collected data from wind tunnel tests in uniform flow*, Tech. rep., Institut fur Schiffbau, Univ. Hamburg

BISHOP, C.M. (2007), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer.

CHAKRABORTY, K.; MEHROTRA, K; MOHAN, C. K.; RANKA S. (1992), *Original contribution: Forecasting the behavior of multivariate time series using neural networks*, Neural Netw., Elsevier Science Ltd.

GHAHRAMANI, Z.; HINTON, G.E. (1996), *Parameter estimation for linear dynamical systems*. Tech. rep., Univ. Toronto

HANSEN, H.; FREUND, M. (2010), *Assistance tools for operational fuel efficiency*, 9th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Budapest, pp.356–366

HOLTROP, J. (1984), *A statistical re-analysis of resistance and propulsion data*. Int. Shipbuilding Progress 31, pp.272–276

HOLTROP, J.; MENNEN, G.G.J (1982), *An approximate power prediction method*, Int. Shipbuilding Progress 29, pp.166–170

JOURNÉE, J. (2003), *Review of the 1985 full-scale calm water performance tests onboard m.v.mighty servant 3,* Tech. Rep. DUT-SHL Report 1361, TU Delft, Ship Hydromechanics Laboratory

JOURNÉE, J.; RIJKE, R.; VERLEG, G. (1987), *Marine performance surveillance with a personal computer*. Tech. Rep. Report 753-P, TU Delft, Ship Hydromechanics Laboratory

LARSEN J. (1993), *Design of neural network filters*, Ph. D. thesis, TU of Denmark

LEIFSSON, L.T.; SÆVARSDÓTTIR, H.; SIGURDSSON, S.T.; VÉSTEINSSON, A. (2008), *Grey-box modeling of an ocean vessel for operational optimization.* Simulation Modelling Practice and Theory 16(8), pp.923–932

MOLLAND, A.F. (2008), *The maritime engineering reference handbook*. Butterworth-Heinemann

PEDERSEN, B.P., LARSEN, J. (2009), *Prediction of full-scale propulsion power using artificial neural networks*, 8th Int. Conf. Computer and IT Applications in the Maritime Industries, Budapest, pp.537–550

RUGGIERO, V.; FILARDI, V.; CUCINOTTA, F. (2007), *Mesh size influence in a CFD code on resistance evaluation of a motor yacht*, 6th Int. Conf. Computer and IT Applications in the Maritime Industries, Cortona, pp.458–466

SVARER, C.; HANSEN, L.K.; LARSEN, J. (1993), *On design and evaluation of tapped-delay neural network architectures*, IEEE Int. Conf. on Neural Networks

TOWNSIN, R. (2003), *The ship hull fouling penalty*, Biofouling 19, pp.9–15

WEIGEND, A.S; HUBERMAN, B.A.; RUMELHART, D.E. (1990), *Predicting the future: A connectionist approach*. Int. J. Neural Systems 1/3, pp.193-209

Appendix C

Gaussian Mixture Models for Analyzing Operational Ship Data

This has been made into a technical report, as it is not novel enough by itself to be an article. However it includes some interesting complementary material to the two first papers, and tackles the modeling problem from a different and quite unique angle.

C.1 Abstract

Onboard performance optimization system have become more common, e.g. for optimal trim advisory systems. These systems are usually based upon physical models, data driven statistical models , or a mixture of both. An often neglected part of this system, is the certainty of the advice given. There are several costs associated with giving wrong advice, among other the crew loosing trust in the system, wasted effort, and even poorer performance. This article proposes an novel approach for modelling the certainty. The data density can to a certain extent be used as an alternative measure to the statistical uncertainty, for determining the trust placed in the results obtained by statistical models. This article presents a computational cheap way of approximating the uncertainty in the results obtained from statistical ship propulsion modelling. The GMM are also discussed in context of regression, missing values handling, and data exploration. The article compares the performance of Gaussian Mixture Regression using the conditional distribution derived from the joint GMM distribution to those obtained by using Artificial Neural Networks.

C.2 Introduction

The *Artificial Neural Networks* (ANN) have been successfully used for estimating the propulsion power, main fuel consumption, and ship speed from full scale measurements. [21, 28, 27, 30, 31] In [27] an ANN solution is proposed to determine the fouling penalty on the ship propulsion efficiency. The deterioration in propulsion performance due to hull and propeller fouling is only a few percents, and can therefore be difficult estimate with traditional performance monitoring methods. In [30] the ANN is proposed for use in a onboard trim optimization application for maximum propulsion efficiency. In [21] the ANN model is suggested to be applied in general operational optimization to achieve maximum energy efficiency.

As stated earlier the *Artificial Neural Network* works well for ship propulsion modelling. The reasons for this are likely that it is a good and well-known model, which has been applied in many other fields successfully before. The computational demands are also very reasonable in most cases on a modern computer. The computational demands of ANN are divided up into a cost for

training, and a cost for making predictions. The training cost does no play a critical role in the applications stated earlier. For example a system which can predict the propulsion power of the vessel only needs to be trained once on a data set which consists of many weeks or months of data. This can be done once, onboard or at the office. The prediction cost plays a larger role, because if the model is used in an onboard optimization system, the results are required within a reasonable amount of time. Evaluations of the ANN are cheap computationally. The cost of a prediction depends on the size of the network used, and the computations mostly involve cheap operations.

Artificial Neural Network however has one major drawback in the context of ship propulsion modelling applications. It does not provide any uncertainty measure with the results found. Justifying decisions on these results are therefore problematic.

The Guassian Process has been investigated as an alternative to the ANN [31], as it provides an uncertainty measure with the result. However it does have other drawbacks. The first hurdle is that the GP's non-parametric nature causes problems with large data sets. Training scales poorly in time, $\mathcal{O}(N^3)$, where N is the number of training samples [24]. Memory requirements are also high for Gaussian Processes and scale $\mathcal{O}(N^2)$. Evaluation of the GP scales poorly, because an evaluation of the GP also costs $\mathcal{O}(N^3)$, for each evaluation point. Work has been done improving the scalability of GP, see Ref. [38]. The second problem with GP is that they do not handle data of high dimensionality very well. It therefore does not seem that the Gaussian Process is a good choice for an onboard system, if these problems are not solved in some manner.

The idea that we bring fourth in this article, is to add a computational cheap system to estimate the amount collected data that support queries made to the model, and can therefore be an extension to the systems already used. That is, we assume that for the used operational ship data there exist some relation between the data density and the certainty of the model output. We will look at modelling the data density using Gaussian Mixture Models (GMM), also called a Mixture of Gaussians (MoG).

The uncertainty of a model is governed by these factors amongst others [1]

- Areas without any data
- Areas with much noise
- Model errors

[1] Maybe this should be stated more mathematically.

The applicability of the GMM in representing the probability distributions found in operational ship data depends mainly on two questions. Firstly that the found approximations are powerful enough to represent the class of density functions found in the data set. Secondly, if such a solution, an optimal set of mixture parameters, can be found in a reasonable manner from a finite set of samples. Regarding the first question, the GMM can approximate any probability density function, as it is an universal approximator of densities. This can be shown using the Wiener's theorem of approximation [32]. Similarly some ANN models are universal function approximators. The problem of determining the parameters is more tricky. Here several approaches have been suggested, but we will look at the *Expectation Maximization* (EM) algorithm. It is an iterative approach, which starts with an initial set of K clusters. First the method evaluates the expectation of the log likelihood of the current model, and the calculates a new set of parameters that optimize maximizes the expectation. There exists a great deal of literature describing the EM algorithm [32, 3], and with consideration for the length of the article, a detailed description of the EM algorithm will not be given here.

C.2.1 Generative and discriminative models

In this section the relation between the GMM and ANN models are briefly studied. A discriminative model describes the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ directly - how do the target variables depend on the observations. Artificial Neural Networks and Gaussian Processes are examples of such models. The GMM is a generative model, where the joint distribution is, $p(\mathbf{x}, \mathbf{y})$, is described instead. The conditional distribution can however be obtained from the joint distribution.Intuitively the discriminative model would perform better at predicting $p(\mathbf{y}|\mathbf{x})$, because the objective of its training process it to find this function. The generative model on the other hand is trained to find the best representation for the joint probability $p(\mathbf{x}, \mathbf{y})$, which does not always result in a good representation of the conditional distribution. However for a finite sample sizes, there is a bias-variance tradeoff, and it is less obvious if the generative or discriminative model will perform better. [5]

Assuming that we have found a GMM model which sufficiently describes the joint probability $p(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ are the model input vector and $\mathbf{t}$ the model target vector. The model for regression can be derived from this, using the conditional probability density $p(\mathbf{t}|\mathbf{x})$. The probability density for each input vector $\mathbf{x}$ can also be found by marginalizing out the targets, so that we only have $p(\mathbf{t})$. The relation between $p(\mathbf{x})$ and the model uncertainty depends on several factors as mentioned earlier: Measurement noise, data density, model parameter noise, amongst others. If the model is predicting in an area with little or no

data, it is like to perform poorly due to this. If the model on the other hand is predicting in an area with high data density, then the measurement error are likely to govern the performance of the model. It is possible only to use the same generative model for both $p(\mathbf{t}|\mathbf{x})$ and $p(\mathbf{x})$ with the advantage of keeping things simpler. But one could also use a discriminative model for $p(\mathbf{t}|\mathbf{x})$, as its performance might be better.

Liang and Jordan [22] give an asymptotic analysis of the prediction error of generative versus discriminative learning. Their high level finding is that 1. the generative approach has the best prediction accuracy when the model is well-specified, that is the ground truth is within the model family and 2. discriminative learning works better in the opposite mismatched scenario. Although the GMM is an universal density approximator we cannot expect that the ground truth is well-specified by a finite component GMM and thus in practice we will have some model mismatch. We can therefore expect as we see that the generative model performance slightly worse than the discriminative ANN model.

C.3 Methods

In this section the Gaussian Mixture Model is explained, and the *Expectation Maximization* algorithm used to find the parameters.

C.3.1 The Gaussian Mixture Model

A multivariate normal distribution or multivariate Gaussian distribution is given by Eq. C.1

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\} \tag{C.1}$$

where he distribution is given by its mean, $\boldsymbol{\mu}$, and covariance, $\boldsymbol{\Sigma}$, matrices.

Mixture models provide a method of describing more complex probability distributions, by combining several probability distributions.

The Gaussian mixture distribution Ref. [3] is given by Eq. C.2

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{C.2}$$

Here we have a linear mixture of K Gaussian density functions, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. It is common to call this a GMM of K clusters. The parameters π_k are called mixing coefficients, which must fulfill $\sum_{k=1}^{K} \pi_k = 1$. Given $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \geq 0$ and $p(\mathbf{x}) \geq 0$ we also have that $0 \geq \pi_k \geq 1$.

The conditional $p(\mathbf{x}_A|\mathbf{y}_B)$ and marginal $p(\mathbf{x})_A$ distributions can be derived for the Gaussian distribution analytically, because a GMM is linear mixture of Gaussians. These two derived distributions have some very useful applications, as we will se shortly.

Let $\mathbf{x}_A$ and $\mathbf{x}_B$ be jointly Gaussian vectors, , and given a GMM with the Guassian distributions $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu_k}, \boldsymbol{\Sigma}_k)$ with $\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1}$ [3, 39, 35]

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix}, \qquad \boldsymbol{\mu}_k = \begin{pmatrix} \boldsymbol{\mu}_{kA} \\ \boldsymbol{\mu}_{kB} \end{pmatrix} \tag{C.3}$$

$$\boldsymbol{\Sigma}_k = \begin{pmatrix} \boldsymbol{\Sigma}_{kAA} & \boldsymbol{\Sigma}_{kAB} \\ \boldsymbol{\Sigma}_{kBA} & \boldsymbol{\Sigma}_{kBB} \end{pmatrix}, \qquad \boldsymbol{\Lambda}_k = \begin{pmatrix} \boldsymbol{\Lambda}_{kAA} & \boldsymbol{\Lambda}_{kAB} \\ \boldsymbol{\Lambda}_{kBA} & \boldsymbol{\Lambda}_{kBB} \end{pmatrix} \tag{C.4}$$

The marginal distribution as

$$p_k(\mathbf{x}_A) = \int p_k(\mathbf{x}) d\mathbf{x}_B = \mathcal{N}(\mathbf{x}_A|\boldsymbol{\mu}_{kA}, \boldsymbol{\Sigma}_{kAA}). \tag{C.5}$$

The conditional distribution for each Gaussian component k is given by Eq. C.6 and C.7, and for the Gaussian Mixture Model in Eq. C.8

$$p_k(\mathbf{x}_A|\mathbf{x}_B) = \frac{p_k(\mathbf{x}_A, \mathbf{x}_B)}{p_k(\mathbf{x}_B)} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{kA|B}, \boldsymbol{\Lambda}_{kAA}{}^{-1}) \tag{C.6}$$

$$\boldsymbol{\mu}_{kA|B} = \boldsymbol{\mu}_{kA} - \boldsymbol{\Lambda}_{kAA}^{-1} \boldsymbol{\Lambda}_{kAB} (\mathbf{x}_B - \boldsymbol{\mu}_{kB}) \tag{C.7}$$

$$p(\mathbf{x}_A|\mathbf{x}_B) = \sum_{k=1}^{K} \pi'_k p_k(\mathbf{x}_A|\mathbf{x}_B), \quad \pi'_k = \frac{\pi_k \mathcal{N}(\mathbf{x}_B|\boldsymbol{\mu}_{kB}, \boldsymbol{\Sigma}_{kBB})}{\sum_k \mathcal{N}(\mathbf{x}_B|\boldsymbol{\mu}_{kB}, \boldsymbol{\Sigma}_{kBB})} \tag{C.8}$$

When using the GMM for regression, the model output can be calculated as the expected value of the GMM probability density function

$$\langle \mathbf{x}_A \rangle = \sum_k \pi'_k \boldsymbol{\mu}_{kA|B} \tag{C.9}$$

The computational costs of evaluating the probability density function form the GMM for a single point is $\mathcal{O}(K)$, where K the number of clusters, evaluations of the probability density function given in Eq. C.1. [2]

C.3.2 GMM parameter estimation

There are several parameters, which need to be estimated for a Gaussian Mixture Model: $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k. In this article the *Expectation-Maximization* (EM) algorithm is used to find the parameters [3, 33]. The EM algorithm is an acknowledged and efficient method for finding the parameters of a GMM. It maximizes the likelihood, but does not ensure that the found solution is a global maximum. Assuming independent and identically distributed samples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. The likelihood for the GMM can then be calculated as

$$\mathcal{L} = \prod_{n}^{N} p(\mathbf{x}_{n=1}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \tag{C.10}$$

The steps of the EM algorithm are given for the GMM according to [3], but for a thorough study of the algorithm there exists a lot of good literature on this subject.

The **expectation step** consist of equation C.11 where the *responsibilities* are calculated from the current parameters

[2] I should investigate this further, because there are some matrix inversions involved here, which depend on the dimensionality of the data.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{C.11}$$

The **maximization step** consist of Eq. C.12, C.13 and C.14,

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{C.12}$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})^{\mathrm{T}} \tag{C.13}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{C.14}$$

The initial cluster centers $\boldsymbol{\mu}_k$ are found using the K-means algorithm. $\boldsymbol{\Sigma}_k$ is set to the covariance of the whole data set, and $\pi_k = 1/K$, so that initially all clusters have the same mixing coefficient.

The appropriate number of clusters K needs to be determined. We have used the Bayesian information criterion (BIC). Having trained a set of models covering a range of K values, either BIC. When the model parameters are found maximizing the likelihood, the likelihood can be increased by adding more clusters, which can result in overfitting. The BIC score avoids the overfitting, because it adds penalty to the number of parameters used in the model, which linked to the number of clusters. Cross-validation could be used as an alternative to the BIC score.

C.4 Experiments

The data used for the experiments is described in [31, 30]. It is a publicly available data set of collected full-scale operational ship data. Briefly described, the system is data driven trim optimization tool, which the crew can consult for optimal trim setting. The optimization process uses a statistical model based upon operational ship data collected from the ship. The collected data consists of speed, wind, wave, trim, amongst other measurements. Our main motivation is obtain a similar measure to the uncertainty measure given the Gaussian Process without the computational overhead, which can be problematic. The idea

is to use the data density distribution to indicate where the system has learnt. This information can be used by the trim optimization system to determine if the found optimum is based upon enough observations to advice the crew to change the trim of the ship. This is very important system characteristic, because giving wrong advice would result in the crew loosing confidence in the system, and unnecessary trim changes.

Three experiments are carried out here.

- Firstly we examine the performance of the GMM for regression on the same problem where the ANN and GP have been used, to see how the GMM performs, and to see if the confidence obtained by the GMM is comparable that the of the GP. This will give an good idea of how the GMM is able to model the ship data.
- Examine the relationship between the pdf and the certainty of the model.

This approach can be used for both the instantaneous/regression model presented in [31] and the dynamic time-delay neural network approach presented in [30]. The differences between the model approaches is discussed in Section C.2.1. The data used for the experiments is collected from a ferry [30].

C.4.1 Regression

In the article [30] several features are derived from the data set, and used to predict the target values: The speed through water and main fuel consumption.

A total number of 31 feature values are derived, of which two are the target values. An Artificial Neural Network (ANN) is used to learn the mapping between the input feature values and target values. The GMM is trained with all these feature values, and then we use the formulas derived earlier to calculate the conditional distributions for the target values. The mean value of the target values are used as output values for the model. The root mean square error (RMSE) obtained for the Artificial Neural Network where 0.38 kn and 47.2 L/h. For the same problem the obtained results using a GMM where 0.42 kn and 51.8 L/h. The GMM was trained on the same training set as the ANN, and the model with the best BIC score was selected. In this case it was a GMM with 8 clusters, see figure C.1 So the performance of the GMM model is close to that of the ANN in this case.

The GMM can easily handle missing signals. Missing signals can simple be marginalized out, so this is very neatly handled by the GMM. This can be an

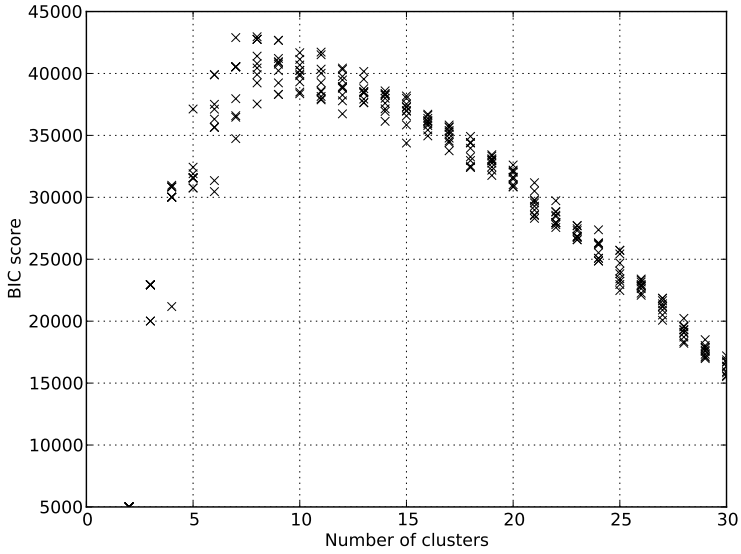

Figure C.1: BIC scores for a range of K clusters. The cluster number giving the best BIC score is $K = 8$.

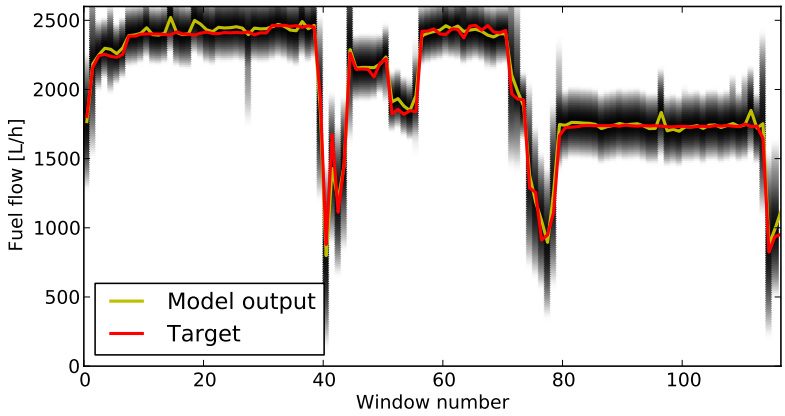

Figure C.2: The output from the GMM model for three selected trips. The same trips have also been used for testing the ANN and GP models before.[31]

advantage for online systems like these, where many signals are collected, and

therefore the chance of one of them missing is higher. Then the system would be able to function to some degree, if the missing signal is not too critical. As an experiment in [31] all features derived from the sea surface level measurements where removed, to see how it will affect the performance of the ANN model. This would require a retraining of the ANN model without these inputs. This is not necessary with the GMM. The obtained RMSE for the GMM is 65.8 L/h compared to 69.1 L/h for the ANN. If only the propeller pitches are retained, then the obtained performance for the GMM is 110.0 L/h RMSE, and 104.0 L/h for the ANN.

If we marginalizing each feature input out one at time, the importance of each input can also be estimated given some test data set. However one might easily get mislead by this method, if there are other signals that contain the same information as the signal left out. For example the propeller pitch is an important feature for predicting the speed of the vessel, but if the ship has two propellers, and the model is presented with two propeller pitch values, then leaving one of them out would not affect the model considerably, because the signals are most likely very similar most of the time. But leaving both of them out, would render the model useless.

Fig. C.3 gives the performance of the model on a subset of the data set. The subset is selected using a cutoff probability density, so that the data points where $p(X_{input}) > p_{cutoff}$ is fulfilled are selected. A whole range of cutoff values are evaluated, and the corresponding regression performance is calculated. This plot can be used to select a reasonable cutoff probability density value, as it gives its relation to the performance, which is more easily interpreted.

C.5 Conclusion

The GMM is viable alternative to the ANN ang GP models. It performs almost as good in the examples using operation ship data for regression presented here. The advantage of the GMM is that it can model the joint probability density function, and that it is possible to find analytic solutions for the marginal and conditional distributions. It is not as computationally demanding as the GP using large data sets.

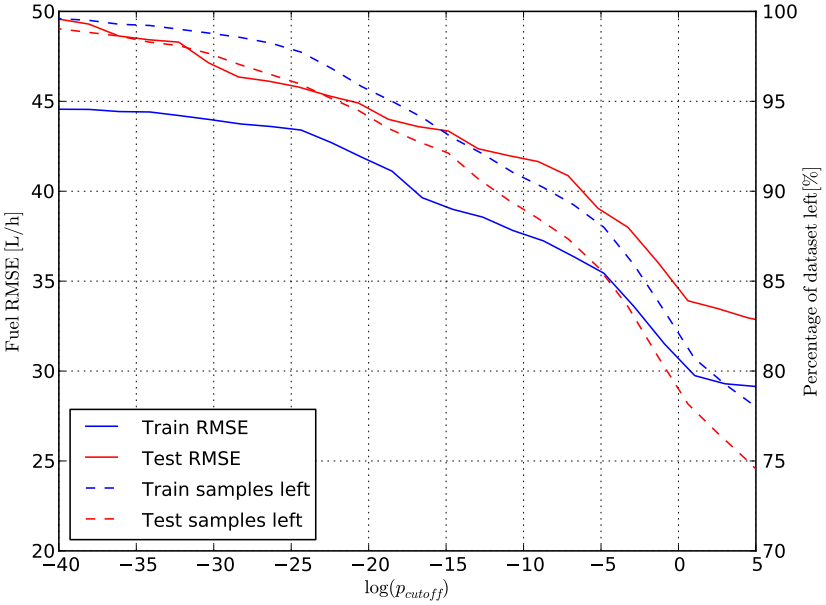

Figure C.3: The regression performance of the model versus the cutoff probability density. The training has been used here.

Bibliography

[1] ADOPT - Advanced Decision support system for ship design, operation, and training.

[2] R. Atwater. Shipboard wave height sensor. Technical report, Ship Structure Committee, 1990.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.

[4] W. Blendermann. Wind loading of ships: Collected data from wind tunnel tests in uniform flow. Technical report, Institut fÃ$\frac{1}{4}$r Schiffbau der Universität Hamburg, 1996.

[5] Guillaume Bouchard and Bill Triggs. The trade-off between generative and discriminative classifiers. In *Proceedings in Computational Statistics, 16th Symposium of IASC*, pages 721–728. Physica-Verlag, 2004.

[6] Michael J. Briggs. Ship squat predictions for ship/tow simulator. Technical report, US Army Corps of Engineers, 2006.

[7] Guangzhi Cao and Charles A. Bouman. Covariance estimation for high dimensional data vectors using the sparse matrix transform. In *Neural Information Processing Systems*, pages 225–232, 2008.

[8] Thomas Eefsen. Optimum trim. *DMI News (Force Institute of Technology*, 2005.

[9] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamic systems. Technical report, Department of Computer Science University of Toronto, 1996.

[10] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical report, Department of Computer Science, University of Toronto, 1996.

[11] H. Hansen and M. Freund. Assistance tools for operational fuel efficiency. In *9th Conf. Computer and IT Applications in the Maritime Industries (COMPIT)*, pages 356–366, 2010.

[12] Mohamad H. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, MA, USA, 1st edition, 1995.

[13] David E. Hess and William E. Faller. Ensuring stability in synergistic computing: Combining flow simulations and neural nets to predict maneuvering. In *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11)*, 2011.

[14] David E. Hess, William E. Faller, Kurt Junghans, Joan L. Lewis, Jonghyuk Lee, and Edward S. Ammeen. Neural network progress at NSWCCD. In *8th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'09)*, 2009.

[15] J. Holtrop. A statistical re-analysis of resistance and propulsion data. *International Shipbuilding Progress*, 31:272–276, 1984.

[16] J. Holtrop and Mennen G.G.J. An approximate power prediction method. *International Shipbuilding Progress*, 29:166–170, 1982.

[17] J.M.J. Journée. Review of the 1985 full-scale calm water performance tests onboard m.v. mighty servant 3. Technical Report DUT-SHL Report 1361, Delft University of Technology, Ship Hydromechanics Laboratory, August 2003.

[18] J.M.J. Journée, R.J. Rijke, and G.J.H. Verleg. Marine performance surveillance with a personal computer. Technical Report Report 753-P, Delft University of Technology, Ship Hydromechanics Laboratory, Mekelweg 2, 2628 CD Delft, The Netherlands, May 1987.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[20] Lena Lastein. Útleiðing av veðurlagsgassi í Føroyum 1990-2001. Technical report, Heilsufrøðiliga Starvsstovan og Oljumálaráðið, 2002.

[21] Leifur TH. Leifsson, Hildur Sævarsdóttir, Sven TH. Sigurdsson, and Ari Vésteinsson. Grey-box modeling of an ocean vessel for operational optimization. *Simulation Modelling Practice and Theory*, 16(8):923–932, 2008.

[22] Percy Liang and Michael I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 584–591, New York, NY, USA, 2008. ACM.

[23] Anthony F. Molland, editor. *The maritime engineering reference handbook*. Butterworth-Heinemann, 2008.

[24] Iain Murray. Gaussian processes and fast matrix-vector multiplies, 2009. Presented at the Numerical Mathematics in Machine Learning workshop at the 26th International Conference on Machine Learning (ICML 2009), Montreal, Canada.

[25] NOAA National Geophysical Data Center. Sunspot numbers. http://www.ngdc.noaa.gov/nndc/struts/results?t=102827&s=5&d=8,430,9, 2010. The International sunspot number is produced by the Solar Influences Data Analysis Center (SIDC), World Data Center for the Sunspot Index, at the Royal Observatory of Belgium.

[26] Francis Noblesse, Dane Hendrix, Lisa Faul, and Jonathan Slutsky. Simple analytical expressions for the height, location, and steepness of a ship bow wave. *Journal of Ship Research*, 50(4):360–370, December 2006.

[27] B. P. Pedersen and J. Larsen. Modeling of ship propulsion performance. In *World Maritime Technology Conference WMTC2009, Jan. 21-24*, Mumbai, India, jan 2009. The Institute of Marine Engineers.

[28] B. P. Pedersen and J. Larsen. Prediction of full-scale propulsion power using artificial neural networks. In *Proceedings of the 8th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'09)*, pages 537–550, may 2009. Budapest, Hungary May 10–12, 2009.

[29] J. P. Petersen. PyPR: Python Pattern recognition. http://pypr.sourceforge.net, 2010–2011.

[30] J. P. Petersen, D. J. Jacobsen, and O. Winther. A machine-learning approach to predict main energy consumption under realistic operational conditions. In *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11)*, pages 305–316, may 2011. Berlin, Germany May 1–4, 2011.

[31] J. P. Petersen, D. J. Jacobsen, and O. Winther. Statistical modelling for ship propulsion efficiency. *Journal of Marine Science and Technology*, 2011. Under review.

[32] K.N. Plataniotis and D. Hatzinakos. *Advanced Signal Processing: Theory and Implementation for Radar, Sonar & Medical Imaging*. CRC Press, 1 edition edition, December 2000.

[33] William H. Press, Saul. A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

[34] Lennart Pundt. Development of an on-board performance and trial trip analysis tool. *Conference on Computer Applications and Information Technology in the Maritime Industrie (COMPIT)*, 2011.

[35] Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[36] Valerio Ruggiero, Vincenzo Filardi, and Filippo Cucinotta. Mesh size influence in a CFD code on resistance evaluation of a motor yacht. In *COMPIT 07 - Computer and IT Applications in the Maritime Industries*, pages 458–466, 2007.

[37] E. Snalson. Flexible and efficient gaussian process models for machine learning, 2007.

[38] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264. MIT press, 2006.

[39] H. G. Sung. *Gaussian Mixture Regression and Classification*. PhD thesis, Rice University, 2004.

[40] R.L. Townsin. The ship hull fouling penalty. *Biofouling*, 19:9–15, 2003.

[41] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.